

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Testování generátorů datového provozu
Testing of Traffic Generators

Zadání bakalářské práce

Student: **Ladislav Pavlacký**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2601R013 Telekomunikační technika

Téma: **Testování generátorů datového provozu**
Testing of Traffic Generators

Zásady pro vypracování:

Cílem bakalářské práce je srovnání různých softwarových generátorů a analyzátorů datového provozu a jejich otestování v různých typech počítačových sítí. Výsledkem práce bude zhodnocení předností a nedostatků jednotlivých generátorů při měření různých přenosových parametrů.

Anotace práce:

1. Srovnajte vlastnosti různých druhů softwarových generátorů a analyzátorů datového provozu. Popište způsob jejich instalace a použití.
2. V laboratorních podmínkách navrhnete a realizujete několik typů počítačových sítí vhodných k testování jednotlivých generátorů.
3. Ve vytvořených sítích změřte různé přenosové parametry generovaných dat při použití různých generátorů.
4. Výsledky srovnajte navzájem a s výsledky získanými při použití hardwarového generátoru.

Seznam doporučené odborné literatury:

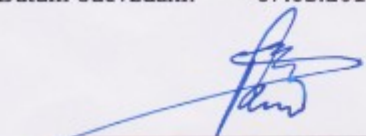
OLIFER, Natalia, OLIFER, Victor. *Computer Networks: Principles, Technologies and Protocols for Network Design*. Chichester: John Wiley & Sons, 2006. ISBN 0470869828.
ODOM, Wendell, CAVANAUGH, Michael. *Cisco QOS Exam Certification Guide*. Indianapolis: Cisco Press, 2004. ISBN 978-1-58720-124-0.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.


Vedoucí bakalářské práce: **Ing. Petr Machník, Ph.D.**

Datum zadání: 16.11.2012

Datum odevzdání: 07.05.2013


prof. RNDr. Vladimír Vašínek, CSc.
vedoucí katedry

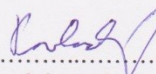



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

Dne: 6.5.2013


.....
podpis studenta

Poděkování

Rád bych poděkoval panu Ing. Petru Machníkovi, Ph.D. za trpělivost, ochotu a odbornou pomoc při konzultacích mé bakalářské práce. Také chci poděkovat za možnost volného přístupu na učebnu N312, kde jsem prováděl všechna měření.

Abstrakt

Tato bakalářská práce se zabývá srovnáním různých softwarových generátorů a analyzátorů datového provozu, jejich instalací v různých operačních systémech a pojednáním o využití těchto nástrojů v praxi. V porovnání se softwarovými generátory a analyzátory datového provozu je vybrán jeden hardwarový nástroj splňující podobný způsob využití. Dále je za pomoci těchto nástrojů provedeno měření základních parametrů kvality datového přenosu, a to na třech typech topologiích. Naměřené hodnoty jsou v závěru práce navzájem porovnány.

Klíčová slova

FETEST ParaScope GigE, Iperf, Thrulay, Qcheck, Propustnost, Zpoždění, Variabilita zpoždění, Ztrátovost paketů, Generátory a analyzátory dat, Topologie

Abstract

This barchelor's thesis follow up the comparison of different software data traffic generators and analyzers , their installation on different operation systems and also reflection about using these tools in practice. In comparison with software data traffic generators and analyzers one hardware tool was selected which meets similar way of usage. Furthermore, the basic parameters measurement of data transmission quality was performed using these tools. It was measured on three topology types. The measured values were compared in the conclusion.

Key words

FETEST ParaScope GigE, Iperf, Thrulay, Qcheck, Throughput, Delay, Jitter, Packet loss, Generators and data analyzers, Topology

Seznam použitých zkratek

Zkratka	Anglický význam	Český význam
IPv4	Internet Protocol version 4	Internetový protokol verze 4
VoIP	Voice over Internet Protocol	Hlas přes internetový protokol
TCP	Transmission Control Protocol	Vysílací kontrolní protokol
UDP	User Datagram Protocol	Protokol pro uživatelské datagramy
ICMP	Internet Control Message Protocol	Kontrolní internetová zpráva
IP	Internet Protocol	Internetový protokol
WiFi	Wireless Fidelity	Bezdrátová technologie
MSS	Maximum Segment Size	Maximální velikost segmentu
MTU	Maximum Transmission Unit	Maximální přenosová jednotka
QoS	Quality of Service	Rezervace a řízení datového toku
UTP	Unshielded Twisted Pair	Nechráněná kroucená dvojlinka
API	Application Programming Interface	Rozhraní pro programování aplikací
OS	Operating System	Operační systém
PC	Personal Computer	Osobní počítač
Handshake	Way to establish a connection	Způsob navázání spojení
802.11g	Wireless Standard in 2,4GHz band with 54Mb/s bit rate.	Bezdrátový standard v pásmu 2,4GHz s maximální rychlostí 54Mb/s.
RFC2544	Benchmarking Methodology for Network Interconnect Devices	Metodika zátěžových testů mezi zařízeními propojenými v síti

Obsah

1	Úvod.....	1
2	QoS.....	2
3	Síťové protokoly.....	3
	3.1 Protokol TCP.....	3
	3.2 Protokol UDP.....	4
	3.3 Protokol ICMP.....	5
	3.4 Protokol IPv4.....	5
4	Základní testy pro měření vlastností sítí.....	6
	4.1 Propustnost.....	6
	4.2 Zpoždění.....	6
	4.3 Variabilita zpoždění (Jitter).....	6
	4.4 Ztrátovost paketů.....	6
5	Hardwarový a softwarové testovací nástroje.....	7
	5.1 FETEST ParaScope GigE.....	7
	5.1.1 Měření propustnosti pomocí Traffic Testu.....	8
	5.1.2 Měření zpoždění a ztrátovosti paketů testem Loopback Delay.....	9
	5.2 Funkce vztahu klient-server.....	10
	5.3 Iperf.....	10
	5.3.1 Popis.....	10
	5.3.2 Ukázka nastavení:.....	13
	5.3.3 Instalace.....	14
	5.4 Thrulay.....	15
	5.4.1 Popis.....	15
	5.4.2 Ukázka nastavení:.....	16
	5.4.3 Instalace.....	19
	5.5 QCheck.....	20
	5.5.1 Popis.....	20
	5.5.2 Ukázka nastavení:.....	22
	5.5.3 Instalace.....	25
6	Návrh topologií.....	29
	6.1 Topologie 1 - technologie Ethernet.....	29
	6.2 Topologie 2 - technologie Ethernet a sériová linka.....	30
	6.3 Topologie 3 - technologie Ethernet a WiFi.....	31

7	Naměřené hodnoty	32
7.1	Topologie 1.....	32
7.2	Topologie 2.....	33
7.3	Topologie 3.....	34
8	Zhodnocení.....	35
8.1	Použité nástroje	35
8.1.1	Shrnutí	37
8.2	Přehled softwarových alternativ.....	37
9	Závěr.....	38
	Použitá literatura	39
	Příloha	xl

1 Úvod

V první kapitole se věnuji problematice zahlcení komunikačních spojů, kdy u některých služeb může dojít k špatnému přenosu paketů díky propustnosti, zpoždění, kolísání zpoždění a ztrátovosti. V této kapitole jsou také vysvětleny metody, kterými se těmto problémům lze vyhnout.

Druhá kapitola souhrnně popisuje protokoly, které jsou základem pro přenos dat v sítích a bez kterých by samotné měření nebylo možné realizovat.

Třetí kapitola pojednává podrobněji o základních testech pro měření vlastností sítí, jakými jsou propustnost, zpoždění, variabilita zpoždění a ztrátovost paketů. Tyto testy budou využity pro naměření hodnot v závěrečné praktické části.

Čtvrtá kapitola zahrnuje v první části popis, funkci, způsob měření a náhled hardwarového generátoru a analyzátoru datového provozu Fetest ParaScope GigE. V druhé části se věnuji popisu, funkci, názorné ukázce možného nastavení a samotné instalaci tří typů softwarových generátorů a analyzátorů dat Iperf, Thrulay a QCheck na různých typech operačních systémů.

Pátá kapitola obsahuje návrh a následně praktickou realizaci tří typů topologií založenou na konfiguraci Cisco zařízení jakými jsou přepínač, směrovač a přístupový bod. První topologie kombinuje směrovače a přepínače s technologií ethernet. Druhá topologie kombinuje směrovače komunikující mezi sebou pomocí sériové linky a přepínače s technologií ethernet. Třetí topologie sestává ze směrovačů, přepínačů a přístupových bodů v režimu bezdrátový most založeném na standardu 802.11g. Všechny tyto tři topologie jsou na svých koncích připojeny k PC stanicím, na kterých jsou nainstalovány a spuštěny softwarové analyzátory a generátory dat. Výjimku tvoří hardwarový generátor a analyzátor dat, který je nastaven v režimu Local Aux Port a tímto tedy dokáže nahradit 2PC stanice svými dvěma ethernetovými porty. Porty se dělí na MAIN port, tedy port, který generuje a vysílá provoz do sítě a AUX port, který generovaný provoz zachytává a vyhodnocuje.

Šestá kapitola se zabývá již samotným testováním generátorů a analyzátorů dat v jednotlivých topologiích. Jedná se o praktickou část s jednotlivými naměřenými výsledky a k nim přiřazeným závěrem.

Sedmá kapitola definuje podmínky provedených měření a nastavených hodnot pro všechny druhy testování. Také zde naleznete zhodnocení a porovnání všech použitých generátorů a analyzátorů dat rozdělené na klady a zápory. Následuje shrnutí a výběr favorita testování. Na závěr této kapitoly naleznete tabulkový přehled softwarových řešení k nástrojům Iperf, Thrulay a QCheck.

V osmé a poslední kapitole se ve své bakalářské práci zaměřuji na porovnání naměřených parametrů pro všechny již uvedené nástroje a to pro každou ze tří topologií.

2 QoS

QoS [1][16] je používáný v telekomunikačních a počítačových sítích tam, kde se k přenosu používá přepojování paketů. Jeho hlavním úkolem je zabránění zahlcení komunikačních spojů a přepojovacích prvků daty. U provozování některých služeb může dojít k špatnému přenosu paketů způsobeným ztrátovostí, zpožděním, kolísáním zpoždění, propustností ad. A právě tyto komplikace QoS řeší.

Podstata se skrývá ve změně priorit datových toků - zvýšení priority jednoho a naopak omezení priorit ostatních, přičemž omezení spočívá v zahazování paketů, jelikož na něj mají povinnost síťové aplikace reagovat snížením přenosové rychlosti.

S pomocí QoS můžeme např. nastavit minimální a maximální přenosové pásmo pro určitá data, nastavit prioritu provozu dle naší potřeby, či rozdělit provoz dat do kategorií.

V dnešní době jsou využívány 3 základní metody QoS:

- **Best-effort services** – mechanismus má QoS nastaven na 0 a snaží se každý paket v co nejkratším čase přenést k cíli.
- **Differentiated services** – pakety se rozdělí do kategorií, což se zaznamenává do jejich hlavičky a poté se s nimi pracuje dle předem definovaných parametrů.
- **Integrated services** - aplikace informují síť o svých požadavcích na kvalitu služby a ta podle požadavku aplikace ověří, zda může mít k dispozici dostatek prostředků na splnění daného požadavku od aplikace a potom se rozhodne, jestli požadavkům vyhoví nebo ne.

3 Sít'ové protokoly

3.1 Protokol TCP

Pro rozšíření vlastností protokolu IP o spolehlivé doručování dat mezi aplikacemi připojených do intranetové, nebo internetové sítě je používán Transmission Control Protocol [14][15]. Tento protokol zprostředkovává zpracování příchozích dat ve správném pořadí a jejich kontrolu.

Data určená k přenosu, přes sít' jsou fragmentována do jednotek určité velikosti, jenž se nazývají pakety (balíčky). Takto vzniklé pakety jsou poté doplněny o hlavičku, obsahující informace o odesílateli, příjemci ad., a připraveny k odeslání.

TCP spojení

Před samotným započítím přenosu datového toku TCP protokol vyžaduje navázání spojení. To je učiněno pomocí předávání speciálních zpráv mezi odesílatelem a příjemcem. Stejným způsobem je nutné, po dokončení celého procesu přenosu, spojení ukončit.

Fáze spojení

V následujících bodech jsou uvedeny 3 hlavní fáze TCP spojení.

1.) Navázání spojení

Aby se mohla vysílat data pomocí TCP protokolu, je potřeba nejprve navázat spojení. Pro tento účel se používá tzv. třicestný handshake. Během navazování spojení se obě koncová zařízení dohodnou na čísla sekvence uváděném v TCP hlavičce a poté je poslán TCP segment, který může mít nastaveno až 8 příznaků (CWR, ECE, URG, ACK, PSH, RST, SYN, FIN).

- I.** klientem je poslán paket se SYN příznakem s uvedeným číslem sekvence (x), číslo odpovědi 0.
- II.** druhá strana si uloží číslo sekvence (x) a odpoví paketem s příznaky SYN-ACK. Jako číslo sekvence nastaví své číslo (y) a do čísla odpovědi vloží (x+1), tzn. další očekávanou hodnotu.
- III.**

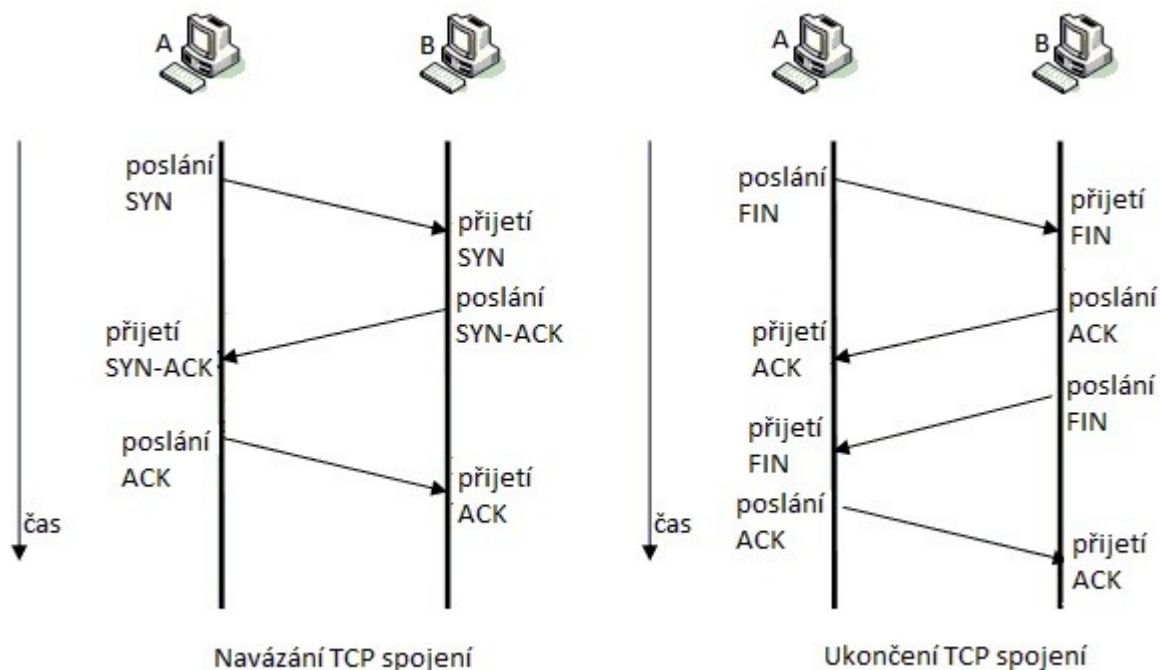
2.) Přenos dat

Realizace přenosu dat s následujícími vlastnostmi:

- Bezchybnost přenosu
- Přijímání paketů ve správném pořadí
- Opětovné poslání ztracených paketů
- Vyřazení duplicitních paketů
- Kontrolu zahlcení

3.) Ukončení spojení

Princip ukončení spojení je obdobný jako při jeho navazování. Nejčastěji je použit čtyřcestný handshake, kdy každá strana samostatně uzavře spojení. Je zde použita sekvence FIN s odpovědí ACK.



Obrázek 4 .1: *Protokol TCP*

3.2 Protokol UDP

Protokol zajišťující menší režii provozu než výše popsany TCP je User Datagram Protocol [14][15]. Ten nezaručuje spolehlivé doručení paketů ve správném pořadí (pouze přidává do hlavičky kontrolní součet) a je tak vhodný pro nasazení, kde se vyžaduje jednoduchost nebo pro aplikace založené na systému request - response. Jeho funkce je využívána v případech, kde se počítá se ztrátami datagramů a není přípustné, aby se ztrácel čas znovu posíláním nedoručených zpráv. Příkladem mohou být video datové toky, internetová rádia a další množství služeb.

UDP protokol je také využíván velmi důležitými službami jako například DNS, DHCP, RIP.

3.3 Protokol ICMP

Internet Control Message Protocol [14][15] je protokol využívaný pro odesílání chybových hlášení, jejichž důvodem může být například nedostupnost služby, stanice nebo směrovače. Tímto však není zajištěna spolehlivost IP protokolu.

ICMP zprávy se typicky vytváří nad IP vrstvou, z datagramu, který reakci vyvolal s tím, že je zapouzdřen s novou IP hlavičkou a připraven k odeslání.

Síťové aplikace však tento protokol samy nevyužívají, čímž se zásadně liší od TCP a UDP.

3.4 Protokol IPv4

IPv4 [14][15] základní protokol používaný v počítačových sítích a internetu je Internet Protocol. Je zodpovědný za směrování datagramů od zdrojového zařízení k cílovému, na základě IP adres obsažených v jejich záhlaví. Je nespolehlivý, jinak řečeno negarantuje doručení, zachování pořadí ani vyloučení duplicit. Tuto spolehlivost obstarávají protokoly vyšších vrstev. Bloky dat posílané v rámci IP sítě se nazývají datagramy.

V dnešní době stále využívaná čtvrtá verze Internetového protokolu (IPv4) se postupně nahrazuje vyšší verzí IPv6, z důvodu vyčerpání IP adres.

IP adresy jsou jednoznačné a jedinečné identifikátory zařízení v síti.

4 Základní testy pro měření vlastností sítí

4.1 Propustnost

Propustnost [1][16] je maximální přenosová rychlost, při které žádný z vyslaných rámců není zahozen. Měření je uskutečněno odesláním určitého počtu rámců testovanému zařízení a následném zjištění kolik jich bylo tímto zařízením předáno dále. Jinými slovy se jedná o rychlost, kdy počet vyslaných rámců přes testované zařízení je roven počtu přijatých rámců. Propustnost je poté označena jako nejvyšší naměřená přenosová rychlost rámců, při které nebyl žádný z nich zahozen a je udávána v počtu přenesených bitů nebo rámců za sekundu.

4.2 Zpoždění

Obecně lze zpoždění [1][16] popsat jako interval mezi okamžikem, kdy byla provedena určitá akce a chvílí, kdy je efekt této akce detekovatelný. V telekomunikacích jej můžeme charakterizovat jako dobu, která uplyne od odeslání zprávy od zdroje, až po její přijetí adresátem. V paketových komunikacích má proměnlivý charakter a ovlivňuje celkové zpoždění přenášené zprávy.

Vzniká v koncových (VoIP telefony, videotelefony, kamery, IP televize atd.) nebo mezilehlých zařízeních (opakovač, rozbočovač, přepínač, most, směrovač) sítě.

Hraje významnou roli u přenosu zvuku, obrazu a dalších multimédií.

Rozlišujeme 2 typy zpoždění. Jednosměrné, představující dobu od odeslání zprávy zdrojem, až po chvíli přijetí a obousměrné, u nějž se počítá čas přenosu zprávy tam i zpět a následné zpracování příjemcem.

4.3 Variabilita zpoždění (Jitter)

Jedná se o kolísání zpoždění [1] přenášených paketů při průchodu sítí. Jinak řečeno je to rozdíl v intervalech mezi přijímanými pakety. Jeho odstranění spočívá v použití vyrovnávací paměti, která tyto rozdíly odbourává.

4.4 Ztrátovost paketů

Ztrátovost paketů [1][16] je počet rámců, které nejsou při konstantní zátěži přeneseny z důvodů nedostatku zdrojů. Testování ztrátovosti rámců je prováděno pomocí škály vstupních přenosových rychlostí a velikostí rámců. Je odstartováno odesláním určitého počtu rámců testovanému zařízení s tím, že je zachována konstantní rychlost přenosu. Je nutné vědět, kolik rámců je testovaným zařízením přeneseno úspěšně. Ztrátovost uváděná v procentech je definována vztahem:

5 Hardwarový a softwarové testovací nástroje

5.1 FETEST ParaScope GigE

ParaScope GigE je vhodným hardwarovým nástrojem pro použití při instalacích, údržbě anebo při odstraňování potíží v lokálních i v metropolitních sítích. Předností je velký barevný displej, malá velikost, hmotnost a dostatečná výdrž baterie, což technikům umožňuje přístroj velmi efektivně využívat i při velmi náročných podmínkách. Uživatel je chopen provádět testování propustnosti, zpoždění, ztrátovosti paketů, síly optického signálu aj. v sítích založených na technologii Ethernet. Tester nabízí dva druhy portů pro metalické vedení a pro optická vedení 10/100/1000 Mb/s. Měření lze také provádět pomocí dvou těchto nástrojů na větší vzdálenosti a vyhodnotit tak naměřené hodnoty mezi sebou. Ovládání nástroje je velmi jednoduché pomocí kláves F1-F4 a směrových šipek.



Obrázek 4.2: Ukázka nástroje FETEST ParaScope GigE

MAIN PORT – port RJ-45, generuje datový provoz

AUX PORT – port RJ-45, přijímá anebo generuje datový provoz

Vlastnosti:

- Generování více datových toků současně
- Práce v jednoportovém, dvouportovém, průchozím režimu a funkci smyčky
- Dva porty metalické 10/100/1000Base-T a dva optické 100/1000Base-X
- Testování dle doporučení RFC 2544
- Měření a vyhodnocení parametrů na L2 až L3, možnost nastavení filtrů
- Testování BER včetně vkládání chyb
- VLAN, VLAN Q-in-Q
- Generování a zpětná analýza několika toků současně pro ověření QoS
- Testy kabelových párů pro vyloučení zkratů, otevřených vedení
- Měření optického výkonu na optických portech
- Export výsledků přes integrovaný USB port
- Rozměry přístroje (V x Š x H) : 190 x 108 x 50 mm
- Váha pouhých 0,42kg

5.1.1 Měření propustnosti pomocí Traffic Testu



Obrázek 4.3: *Test propustnosti*

Nastavení:

IP adresa MAIN PORT (klient)	192.168.1.2
IP adresa AUX PORT(server)	192.168.2.2

5.1.2 Měření zpoždění a ztrátovosti paketů testem Loopback Delay



Obrázek 4 .4: *Test zpoždění a ztrátovosti paketů*

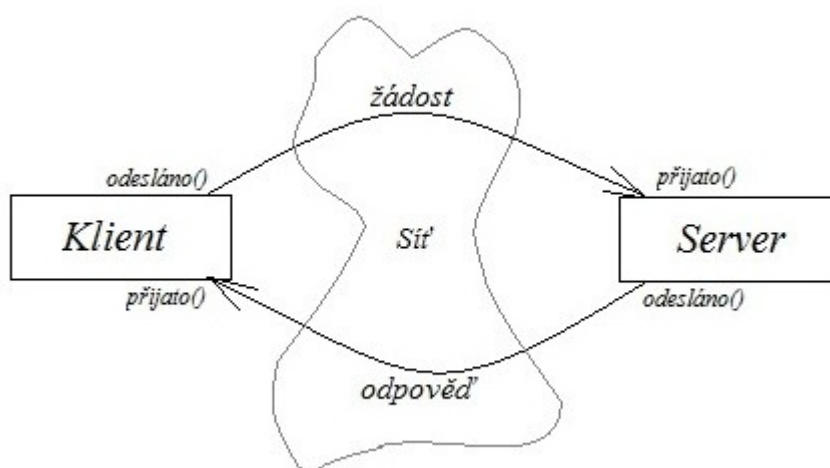
Nastavení:

IP adresa MAIN PORT	192.168.1.2
IP adresa AUX PORT	192.168.2.2

Doba testování nastavena pro každé měření na 10 sekund. Generované přenosové pásmo 1Mb/s. Velikost rámce 1500B. Test probíhá na vrstvě IP protokolu.

5.2 Funkce vztahu klient-server

Klient-server popisuje vztah mezi dvěma počítačovými programy (v mé bakalářské práci jde o software Iperf, Thrulay a QCheck), v nichž první program, klient, žádá o služby jiný program zvaný server. Na základě tohoto vztahu probíhá měření vlastností sítí jako je propustnost, zpoždění, ztrátovost paketů a variabilita zpoždění.



Obrázek 4 .5: *Funkce vztah klient-server*

5.3 Iperf

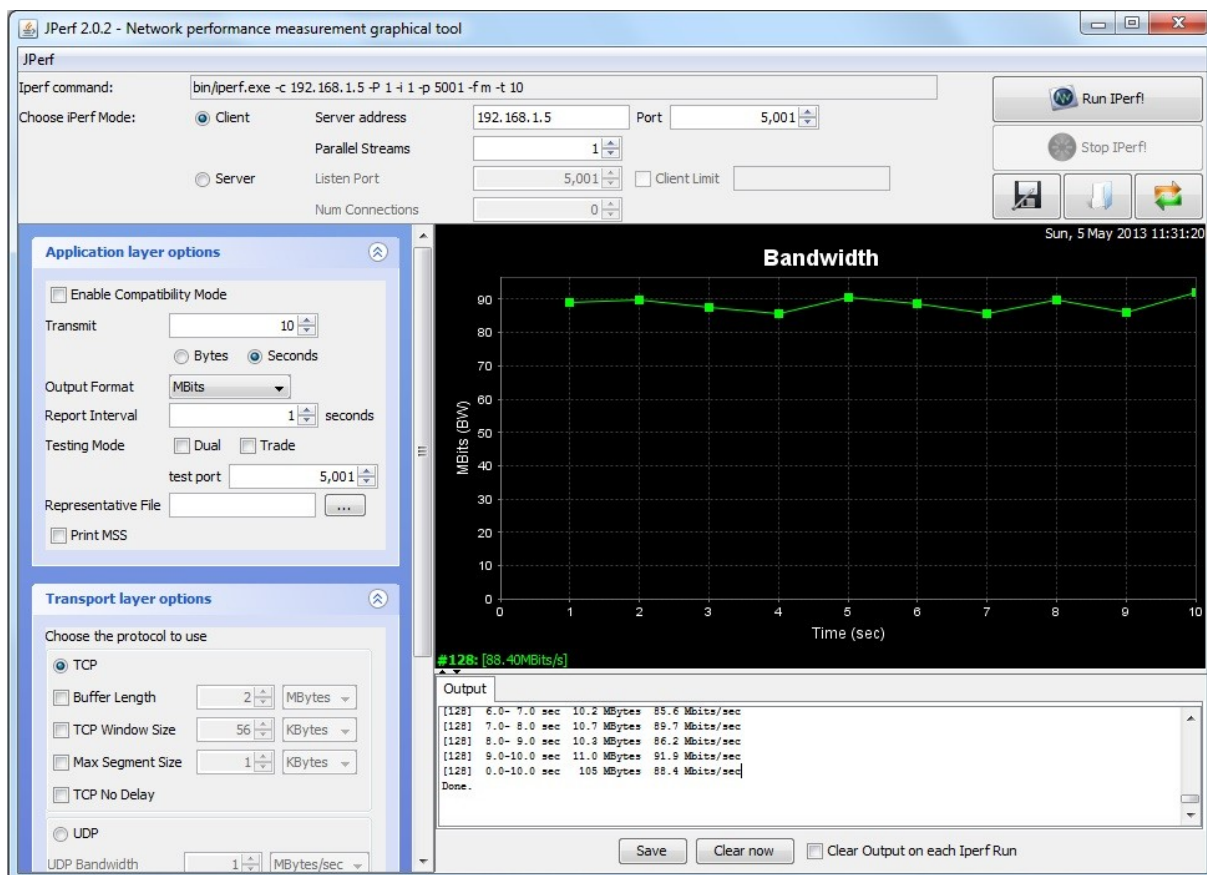
5.3.1 Popis

Diagnostický nástroj sloužící k testování propustnosti datové sítě, který může vytvořit TCP a UDP datové proudy a změřit tak propustnost sítě. Iperf [4] byl vyvinut společností Distributed Applications Support Team (DAST) v národní laboratoři pro aplikovaný výzkum sítí (National Laboratory for Applied Network Research – NLANR) na území USA. Program Iperf umožňuje nastavit různé parametry měření, které mohou být použity pro testování sítě nebo jako náhradní způsob pro optimalizaci a ladění sítě. Propustnost mezi klientem a serverem může být jednosměrný nebo obousměrný test. Velké využití má hlavně díky tomu, že jde o multiplatformní nástroj a lze ho tedy spustit přes libovolnou síť. Můžeme tak měřit a srovnávat výsledky na metalických, či bezdrátových sítích a to objektivně. Metodika kontroly je mezi různými operačními systémy stejná tzn. Stejná typologie při volení testu v příkazové řádce s výjimkou pro operační systém Microsoft Windows, pro který je volně dostupná grafická nástavba naprogramovaná v jazyce Java, která disponuje názornými grafickými výstupy pro naměřené hodnoty. Při testování s Iperf můžeme využít testování za pomoci UDP anebo TCP protokolu. Při využití testování UDP metodou umožňuje uživateli určit velikost datagramu a poskytuje výsledky pro propustnost datagramu a také ztrátovost paketů. TCP testování měří užitečné zatížení sítě, tedy propustnost. Nastavení Iperf se provádí v příkazovém řádku. Nastavení Iperf [7] probíhá v samotné aplikaci. Zobrazení naměřených výsledků lze pomocí volitelné volby zobrazit i na straně serveru. Iperf a Iperf patří mezi otevřený software (open source software – OSS) a spustitelný je na OS Linux, Windows a Unix.

- Měření prováděná na verzi Iperf 2.0.5-2

```
^Croot@ladis-F5RL:/home/ladis# iperf -s
-----
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
-----
[  4] local 192.168.1.5 port 5001 connected with 192.168.1.2 port 56246
[ ID] Interval      Transfer    Bandwidth
[  4]  0.0-10.0 sec  105 MBytes  88.4 Mbits/sec
```

Obrázek 4 .6: Ukázka nástroje Iperf



Obrázek 4 .7: Ukázka nástroje Jperf

Vlastnosti:

- Měření v režimu klient – server.
- TCP
 - Měření propustnosti
 - Zpráva o velikosti MSS/MTU
 - Podpora pro změnu velikosti oken (window size)
 - Podpora více vláken současně. Klient a server může mít více spojení současně
- UDP
 - Klient může vytvořit datový tok UDP zvolením šířky přenosového pásma
 - Měří ztrátovost paketů
 - Měří variabilitu zpoždění
 - Podpora multicast
 - Podpora více vláken současně. Klient a server může mít více spojení současně. Nefunkční v Microsoft Windows
- Nastavení volitelných parametrů a zobrazení výsledků může být upřesněno volbou K (kilobajt) a M (Megabajt). Vhodnější je tedy použití 128K namísto 131072B.
- Lze nastavit dobu generování dat v sekundách
- Podle velikosti vypsání dat zvolí nejlépe se hodící jednotku
- Server dokáže zpracovat více připojení v jednom testu
- Zobrazení propustnosti, variability zpoždění a ztrátovosti paketů ve stanovených intervalech
- Server lze spustit jako aplikace spuštěnou na pozadí anebo ve Windows jako Windows službu
- Lze nastavit velikost paketu
- Server v základním nastavení naslouchá na portu 5001
- Výpis naměřených hodnot na straně klienta a po předvolbě i u serveru zároveň
- Vytížení linky v předem stanovené šířce pásma
- Podpora operačních systémů:
 - Windows 2000, XP, 2003, Vista, 7
 - Linux 32bitů (i386), Linux 64bitů (AMD64)
 - MacOS X
 - Oracle Solaris

5.3.2 Ukázka nastavení:

TCP - Propustnost

- Na straně klienta:

```
#iperf -c 192.168.1.2 -m -i 1 -p 5001 -f m -t 10
```

- Na straně serveru:

```
#iperf -s -i 1 -f m
```

- Popis jednotlivých voleb:

#iperf	název software v příkazové řádce
-c	spuštění jako klient
-s	spuštění jako server
-m	vypíše maximální velikost TCP segmentu (MSS)
-i 1	výpis propustnosti co 1 sekundu
-p 5001	nastavení portu, na kterém server naslouchá
-f m	formát zobrazení výsledků, m pro Mb/s
-t 10	doba trvání testu 10 sekund
192.168.1.2	IP adresa serveru

UDP – Variabilita zpoždění, ztrátovost paketů

- Na straně klienta:

```
iperf -c 192.168.1.2 -u -m -i 1 -p 5001 -f m -b 1.0M -t 10
```

- Na straně serveru:

```
#iperf -s -i 1 -f m
```

- Popis jednotlivých voleb:

#iperf	název software v příkazové řádce
-c	spuštění jako klient
-s	spuštění jako server
-u	použije UDP pakety namísto TCP
-m	vypíše maximální velikost TCP segmentu (MSS)
-i 1	výpis propustnosti co 1 sekundu
-p 5001	nastavení portu, na kterém server naslouchá
-f m	formát zobrazení výsledků, m pro Mb/s
-b 1.0M	nastavení cílové šířky pásma. 1.0M – generování 10 ⁶ b/s
-t 10	doba trvání testu 10 sekund
192.168.1.2	IP adresa serveru

Kompletní přehled voleb

- Manuál dostupný v OS Linux z příkazové řádky: `man iperf`
- Online literatura [4]

5.3.3 Instalace

Ubuntu 12.10:

- Iperf

1) První krok – instalace

Instalace `iperf` je dostupná ihned z příkazové řádky v OS Ubuntu. Pomocí správy balíčků `apt-get`.

```
root@ladis-F5RL:/home/ladis# apt-get install iperf
Reading package lists... Done
```

Obrázek 4 .8: Instalace Iperf

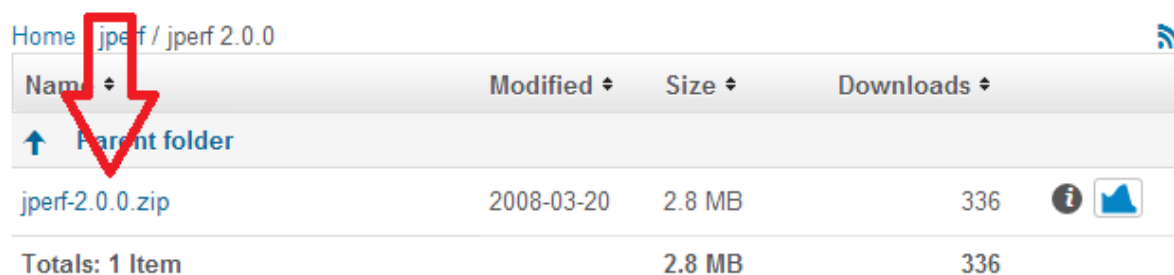
MS Windows 7 32-bit:

- Jperf

1) První krok – stažení aplikace

Internetové stránky repozitáře Sourceforge, literatura [7].

Zde otevřeme adresář `jperf 2.0.0` a stáhneme soubor `jperf-2.0.0.zip`



Name	Modified	Size	Downloads
Parent folder			
jperf-2.0.0.zip	2008-03-20	2.8 MB	336
Totals: 1 Item		2.8 MB	336

Obrázek 4 .9: Stažení aplikace IS

2) Druhý krok – rozbalení a spuštění

rozbalení komprimovaného `jperf` a spuštění dávkového souboru `jperf`

bin	19.3.2008 15:46	Složka souborů	
lib	19.3.2008 15:48	Složka souborů	
jperf	19.3.2008 15:44	Dávkový soubor s...	1 kB
jperf.sh	19.3.2008 15:44	Soubor SH	1 kB
jperf-2.0.0	19.3.2008 15:58	Executable Jar File	61 kB

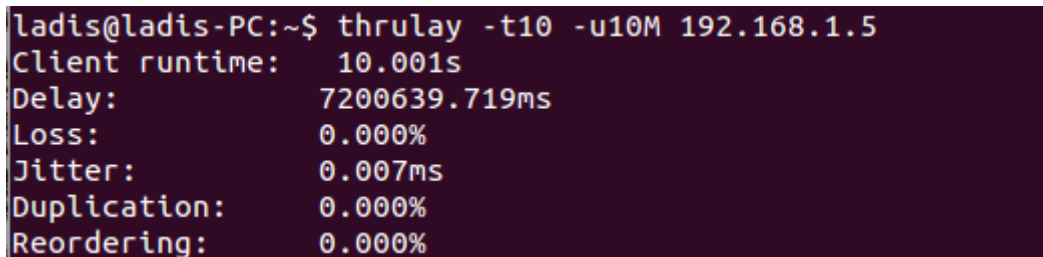
Obrázek 4 .1: Složka Jperf

5.4 Thrulay

5.4.1 Popis

Diagnostický nástroj vytvořen Stanislavem Shalunovem. Původní verze nese název i2perf. Díky podobě názvu s nástrojem Iperf byl software přejmenován na konečnou verzi a to Thrulay [8][9]. Od Thrulay jsou také odvozeny verze s názvy Thrulay-ng a Thrulay-hd, které se liší úpravami při testování a jsou vytvořeny vývojáři mimo původního autora, tedy Stanislava Shalunova. Thrulay slouží k testování propustnosti datové sítě, který může vytvořit TCP a UDP datové proudy a změřit tak propustnost sítě. Tímto nástrojem lze zjistit propustnost sítě, zpoždění, ztrátu paketů, duplikaci paketů a doručení paketů mimo pořadí. Thrulay patří mezi otevřený software (open source software – OSS), využití je jen v operačním systému Linux.

- Měření prováděná na verzi Thrulay 0.9
 - V této verzi jsou již převzaty prvky z Thrulay-ng, Thrulay-hd



```
ladis@ladis-PC:~$ thrulay -t10 -u10M 192.168.1.5
Client runtime: 10.001s
Delay: 7200639.719ms
Loss: 0.000%
Jitter: 0.007ms
Duplication: 0.000%
Reordering: 0.000%
```

Obrázek 4 .8: Ukázka nástroje Thrulay

Vlastnosti Thrulay:

- Měření v režimu klient – server.
- TCP
 - Měří oboustranné zpoždění
 - Měří propustnost
 - Podpora pro změnu velikosti oken (window size)
 - Podpora více vláken současně. Klient a server může mít více spojení současně
- UDP
 - Klient může vytvořit datový tok UDP zvolením šířky přenosového pásma
 - Měří ztrátovost paketů
 - Měří pořadí doručených paketů
 - Měří jednosměrné zpoždění
 - Podpora více vláken současně. Klient a server může mít více spojení současně. Nepodporováno v OS Microsoft Windows
 - Vytížení linky v předem stanovené šířce pásma

- Nastavení volitelných parametrů a zobrazení výsledků může být upřesněno volbou k (kilobit) a M (Megabit).
- Lze nastavit dobu generování dat v sekundách
- Zobrazení naměřených hodnot na obrazovce je přehledné
- Server dokáže zpracovat více připojení v jednom testu
- Zobrazení propustnosti, variability zpoždění a ztrátovosti paketů ve stanovených intervalech
- Server je spouštěn jako démon
- Lze nastavit velikost paketu
- Server v základním nastavení naslouchá na portu 5003
- Výpis naměřených hodnot pouze na straně klienta
- Podpora operačních systémů:
 - Linux 32bitů (i386), Linux 64bitů (i686)
 - MacOS X
 - Oracle Solaris
 - BSD (FreeBSD, OpenBSD, NetBSD)

5.4.2 Ukázka nastavení:

TCP – Propustnost, zpoždění

- Na straně klienta:


```
#thrulay -i1 -t10 192.168.1.2
```
- Na straně serveru:


```
#thrulayd
```
- Popis jednotlivých voleb:

#thrulay	spuštění jako klient
#thrulayd	spuštění jako server
-i1	výpis propustnosti co 1 sekundu
-t10	doba trvání testu 10 sekund
192.168.1.2	IP adresa serveru

UDP – Variabilita zpoždění, ztrátovost paketů

- Na straně klienta:

```
# thrulay -t10 -u1M 192.168.1.2
```

- Na straně serveru:

```
#thrulayd
```

- Popis jednotlivých voleb:

#thrulay	spuštění jako klient
#thrulayd	spuštění jako server
-i1	výpis propustnosti co 1 sekundu
-t10	doba trvání testu 10 sekund
-u1M	nastavení šířky generovaných dat. 1M je generování 10 ⁶ b/s
192.168.1.2	IP adresa serveru

- Kompletní přehled voleb

- Manuál dostupný v OS Linux z příkazové řádky:

- Na straně klienta: man thrulay
- Na straně serveru: man thrulayd
- Online literatura [6]

Popis rozšíření základní verze Thrulay:

- Thrulay-ng [10]

Vytvořen Bernhardem Lutzmannem v projektu Google „summer of code“

Je založen na verzi Thrulay 0.6. Sestává z částí na straně klienta (thrulay) a serveru (thrulayd). Server je po spuštění spuštěn jako aplikace na pozadí.

- TCP testování: Podpora více současně běžících vláken, statistika propustnosti, minimální, průměrná, maximální hodnota zpoždění.
- UDP testování: statistika množství duplikací paketů, ztrátovosti paketů, změna pořadí doručených paketů
- Podpora pro Linux, BSD (FreeBSD, OpenBSD, NetBSD), Mac OS X a Oracle Solaris
- Ověřování klienta založeném na síťové masce
- Použití rychlejší gettimeofday() implementace
- Podpora pro IPv6 protokol

- Thrulay-hd [11]

Vytvořen Huadongem Liuem v projektu Google „summer of code“

- Vícenásobný počet vláken jak u TCP, tak u UDP
- API pro programové provedení thrulay testů
- Nezávislost na platformě OS
- Podpora pro IPv6 protokol a ověření klienta pomocí jeho IP adresy

5.4.3 Instalace

Ubuntu 12.10:

1) První krok – stažení aplikace

Internetové stránky společnosti Internet2, literatura [8]. Žlutou šipkou označen odkaz *Downloads*. Na nově otevřeném internetovém odkazu zvolíme *Download thrulay-0.9.tar.gz*. Tímto stáhneme komprimovaný archiv.



Obrázek 4.9: Stažení aplikace

2) Druhý krok – rozbalení archivu

V příkazové řádce se přepneme do místa, kde je umístěn stažený soubor *thrulay-0.9.tar.gz* a ten rozbalíme příkazem:

```
tar xzf /thrulay-0.9.tar.gz
```

3) Třetí krok – kompilace

V příkazové řádce se přepneme do složky, kde jsme archiv rozbalili a zadáme:

```
./configure
make
```

4) Čtvrtý krok – instalace

Pro instalaci zadáme:

```
Sudo make install
```

5) Pátý krok – spuštění

V příkazové řádce stačí již zadat *iperf*, ke čtení dokumentace *man iperf*

5.5 QCheck

5.5.1 Popis

Diagnostický nástroj vytvořen společností NetIQ. QCheck [12][13] slouží k testování propustnosti datové sítě. Může vytvořit TCP a UDP datové proudy a změřit tak propustnost sítě. Tímto nástrojem lze zjistit propustnost sítě, zpoždění, ztrátu paketů a uzly na cestě datagramů od zdroje k cíli. V současnosti je QCheck oficiálně pod správou společnosti Ixia. Jedná se o otevřený software (open source software – OSS), který nese omezení v rozsahu nastavení parametrů jako je počet opakování testů a velikost přenášených dat. QCheck se proto hodí pro měření linek o maximální rychlosti do 100Mb/s. Společnost Ixia nabízí také nástroj IxChariot, který se hodí pro linky větší než 100Mb/s. Je tedy bez jakéhokoliv omezení, a svým zpracováním se tak hodí i pro profesionální nasazení. IxChariot je tedy univerzální nástroj, ale již se nejedná o otevřený software, ale placenou aplikaci.

- Měření prováděná na verzi QCheck 3.0



Obrázek 4.10: Ukázka nástroje QCheck

Vlastnosti:

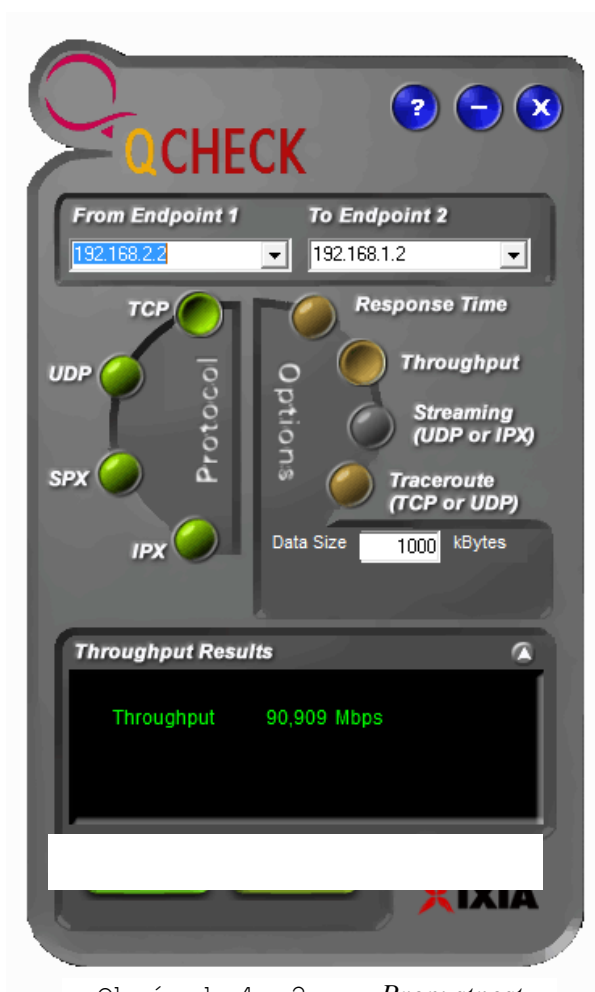
- Měření v režimu klient server
 - OS Windows – klient (QCheck), server (QCheck)
- TCP
 - Měření propustnosti (Omezení - velikost dat 1-1000 kilobajtů)
 - Měření zpoždění (Omezení - počet opakování 1-10, velikost dat 1-1000 kilobajtů)
 - Zjišťování uzlů na dané trase
- UDP
 - Měření propustnosti (Omezení - velikost dat 1-1000 kilobajtů)
 - Měření zpoždění (Omezení - počet opakování 1-10, velikost dat 1-1000 kilobajtů)
 - Vytížení linky v předem stanovené šířce pásma (Omezení - 1kb/s až 1Mb/s)
- Podpora dalších protokolů:
 - SPX (Měření propustnosti, zpoždění)
 - IPX (Měření propustnosti, zpoždění a volby šířky pásma)
- Zobrazení propustnosti, variability zpoždění a ztrátovosti paketů ve stanovených intervalech
- Lze nastavit velikost paketu
- Server v základním nastavení naslouchá na portu 10113
- Výpis naměřených hodnot na straně klienta
- Podpora operačních systémů:
 - OS Windows 2000, NT a XP.
 - Linux, MacOS X, Oracle Solaris – spuštění pouze jako server. Nutno spustit jen koncový bod (Performance Endpoint software)

5.5.2 Ukázka nastavení:

TCP

- Na straně klienta:

From Endpoint 1	IP adresa klienta
To Endpoint 1	IP adresa serveru
TCP + Throughput	Měření TCP propustnosti
Data Size	Velikost generovaných dat

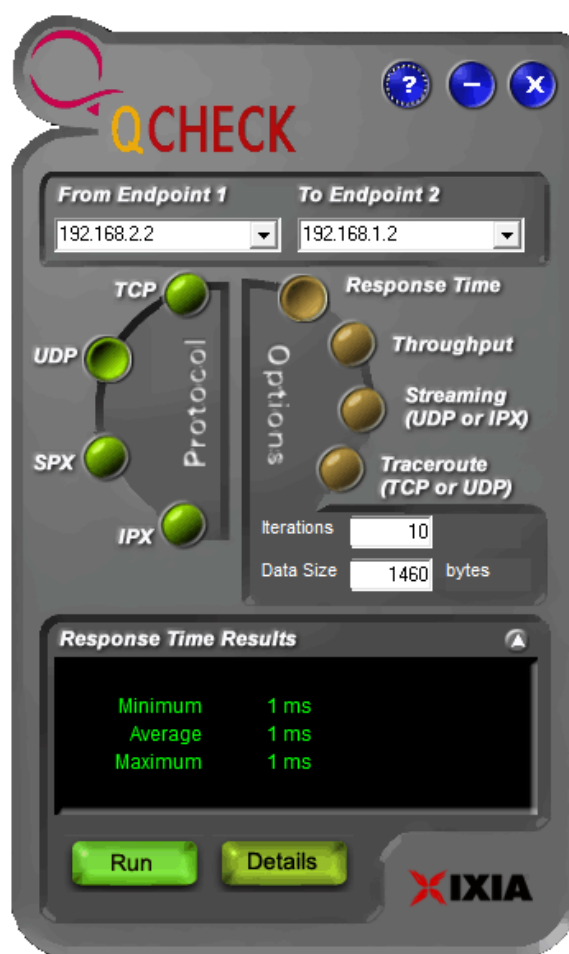


Obrázek 4 .2: Propustnost

UDP

- Na straně klienta:

From Endpoint 1	IP adresa klienta
To Endpoint 1	IP adresa serveru
UDP + Response time	Měření zpoždění
Iteration	Počet opakování
Data Size	Velikost generovaných dat

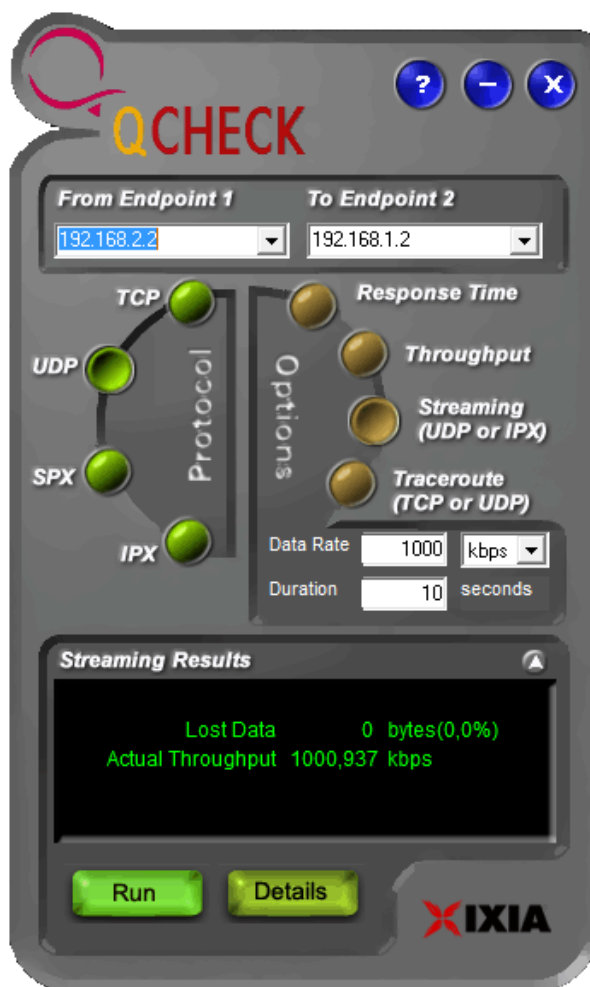


Obrázek 4 .11: Zpoždění

UDP

- Na straně klienta:

From Endpoint 1	IP adresa klienta
To Endpoint 1	IP adresa serveru
UDP + Streaming	Měření Ztrátovosti paketů
Data Rate	Nastavení cílové šířky pásma na 1Mb/s
Duration	Doba trvání testu



Obrázek 4 .3: Ztrátovost paketů

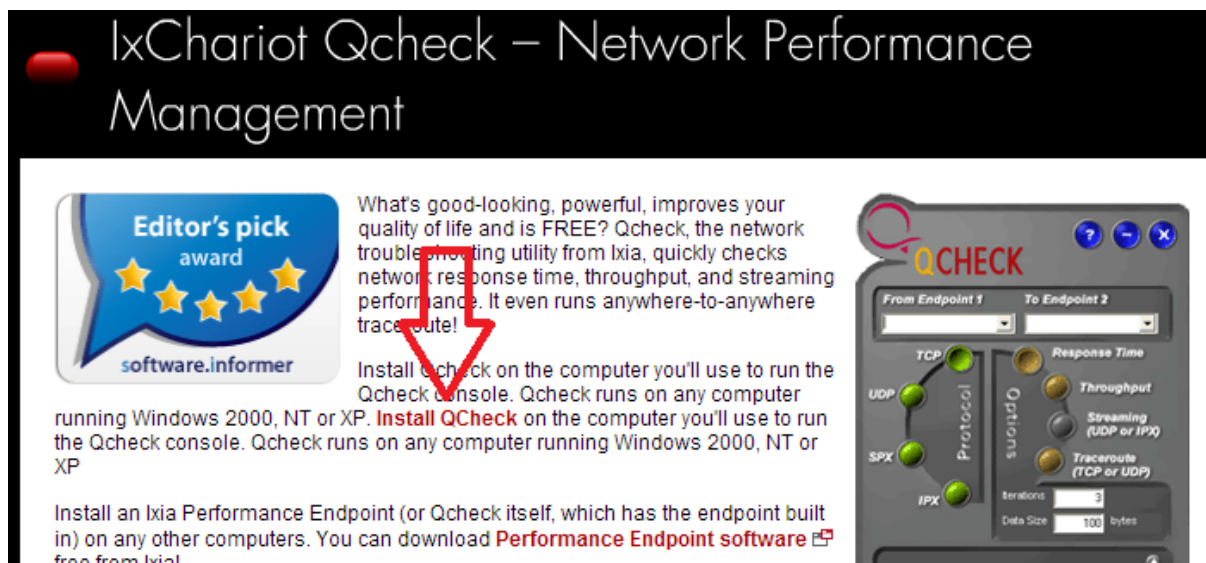
- Na straně serveru:
 - Po instalaci v OS Windows QCheck bez spuštění automaticky naslouchá na portu 10113
- Kompletní přehled voleb
 - Manuál dostupný na panelu QCheck pod symbolem ?
 - Online literatura [12][13]

5.5.3 Instalace

MS Windows 7 32-bit:

1) První krok – stažení aplikace

Internetové stránky společnosti Ixia, literatura [12]. V textu níže odkaz na *Install QCheck*:



Obrázek 5.12: Stažení aplikace

2) Druhý krok – vyplnění formuláře

Budeme automaticky přesměrováni na formulář, kde je nutné vyplnit pole označené hvězdičkou (jméno, příjmení, emailová adresa, telefonní číslo atd.), potvrdíme *Submit*.

Obrázek 5.13: Vyplnění formuláře

3) Třetí krok – vyzvednutí emailu

Na email uvedený v registraci zašle společnost Ixia poděkování za registraci včetně odkazu ke stažení *Download QCheck*. Stisknutím odkazu uložíme instalační soubor na pevný disk.

Dear Karel,

Thank you for your interest in QCheck.

Qcheck is a simple and effective utility for testing network performance, measuring response time, throughput, availability and lost packets.

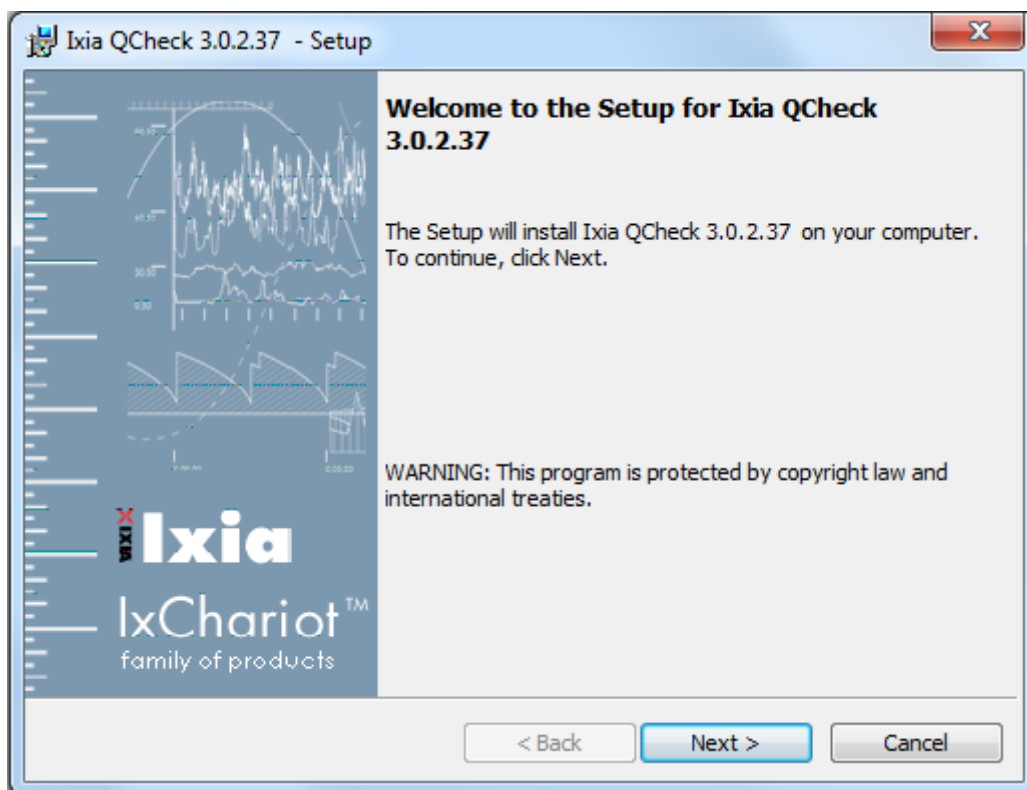
[Download QCheck](#).

If your browser experiences any difficulty in accessing the executable via this link then please save the file locally as "[qcinst3.0.exe](#)" and execute from your machine.

Obrázek 5.14: Vyzvednutí emailu

4) Čtvrtý krok – instalace QCheck

V první fázi po spuštění nás instalací provede pomocný instalační průvodce a poté se teprve spustí instalační průvodce samotného QCheck. Následuje potvrzení licenčního ujednání, výběru místa instalace a instalace. Součástí instalace je i koncový bod (Performance Endpoint software).



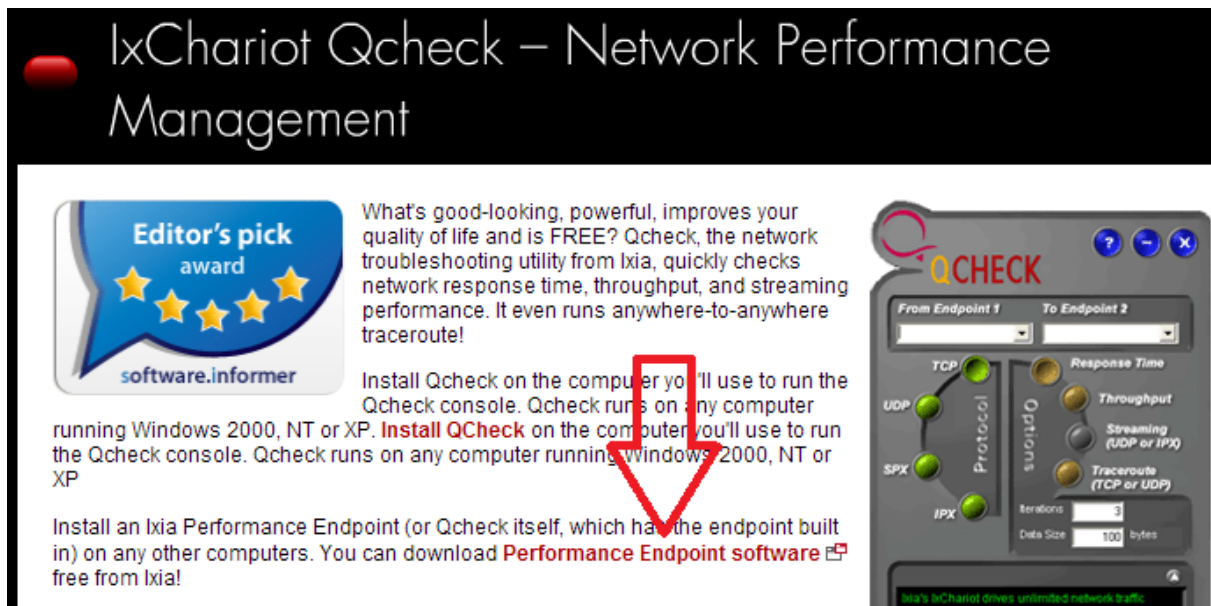
Obrázek 5.15: Instalace QCheck

Instalace koncových bodů (Performance Endpoint software)

Ubuntu 12.10:

1) První krok – repozitář koncových bodů

Internetové stránky společnosti Ixia, literatura [12]. V textu níže odkaz na *Performance Endpoint software*. Odkaz nás odkáže na repozitář koncových bodů pro různé verze OS (Windows XP, Windows Vista, Linux for Itanium, Linux x86, Apple MacOS X, Sun Solaris).



Obrázek 5.16: Repozitář koncových bodů

2) Druhý krok – instalace koncového bodu

Pro ukázkou v repozitáři koncových bodů zvolíme stažení archivu na záložce Linux x86 / 32-Bit (i386) s balíčkovacím systémem Debian.

3) Třetí krok – rozbalení archivu

Pro ukázkou v repozitáři koncových bodů zvolíme stažení archivu na záložce Linux x86 / 32-Bit (i386) a provedeme rozbalení archivu přepnutím se do místa, kde jsme archiv uložili. Poté zadáme:

```
tar xzf /pelnx_730.tar
```

4) Čtvrtý krok – kompilace

V příkazové řádce se přepneme do složky, kde jsme archiv rozbalili a zadáme:

```
./configure  
make
```

5) Pátý krok – instalace

Pro instalaci zadáme:

```
Sudo make install
```

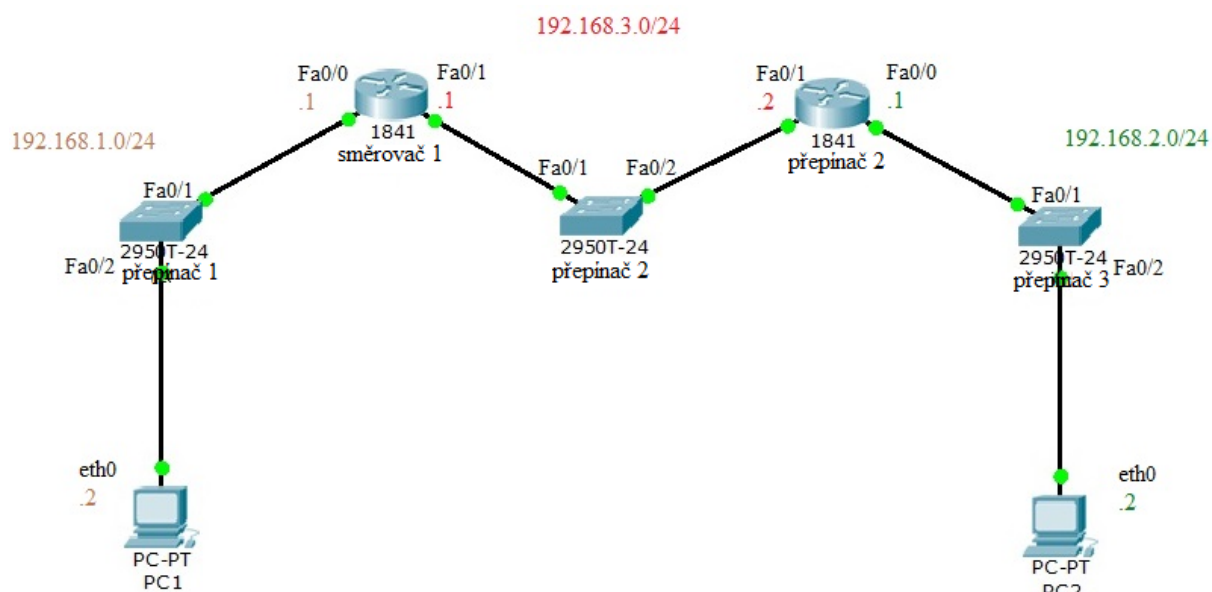
6) Šestý krok – spuštění

Koncový bod je spuštěn na pozadí automaticky při startu OS

6 Návrh topologií

6.1 Topologie 1 - technologie Ethernet

Schéma zapojení:



Obrázek 4.17: Topologie 1, návrh v Cisco Packet Tracer

Popis zapojení:

Na obrázku 4.17 můžeme vidět zapojení laboratorní úlohy založené na technologii ethernet s maximální teoretickou přenosovou rychlostí 100Mb/s. Pro přenos dat mezi propojenými zařízeními je použita kroucená dvojlinka ve variantě UTP.

PC2 je server a PC1 je klient.

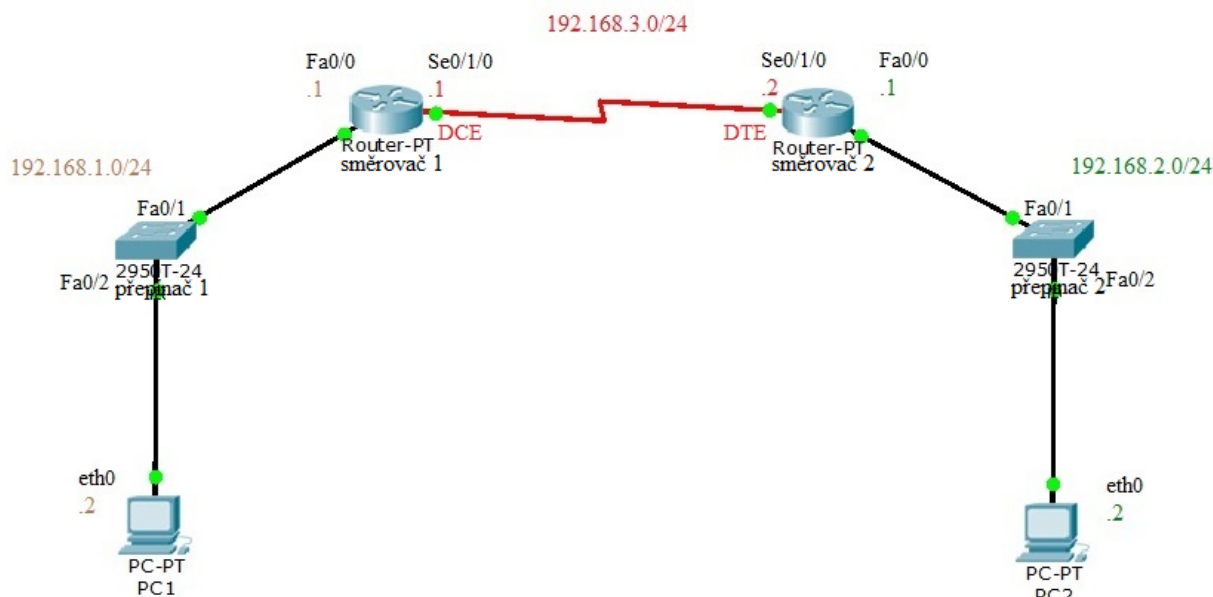
Seznam použitých zařízení:

PC1, PC2	OS Windows 7 32-bit, OS Ubuntu 12.10
Směrovač 1, 2	Cisco 2800 series
Přepínač 1, 2, 3	Cisco Catalyst 2960 series

Obrázek 4.18: Seznam zařízení topologie 1

6.2 Topologie 2 - technologie Ethernet a sériová linka

Schéma zapojení:



Obrázek 4 .19: Topologie 2, návrh v Cisco Packet Tracer

Popis zapojení:

Na obrázku 4.19 můžeme vidět zapojení laboratorní úlohy založené na technologii ethernet a sériovou linku, která je na straně směrovače 1 udává takt 128000bitů za sekundu. Pro přenos signálu mezi zbývajících zařízeními je použita kroucená dvojlinka ve variantě UTP. Rychlost sériového rozhraní na straně směrovače 1 byla nastavena na 128kb/s.

PC2 je server a PC1 je klient.

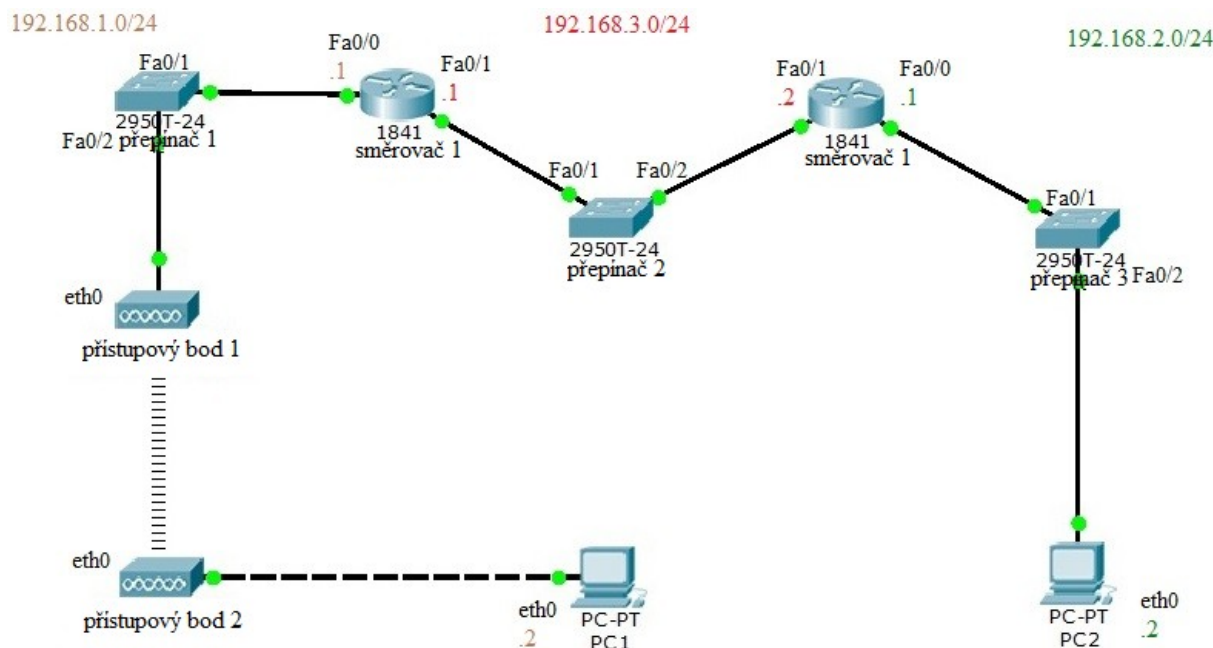
Seznam použitých zařízení:

PC1, PC2	OS Windows 7 32-bit, OS Ubuntu 12.10
Směřovač 1, 2	Cisco 2800 series
Přepínač 1, 2	Cisco Catalyst 2960 series

Obrázek 4 .20: Seznam zařízení topologie 2

6.3 Topologie 3 - technologie Ethernet a WiFi

Schéma zapojení:



Obrázek 4 .21: Topologie 3, návrh v Cisco Packet Tracer

Popis zapojení:

Na obrázku 4.21 lze vidět zapojení laboratorní úlohy s technologií ethernet doplněnou o dva přístupové body nastavené v režimu most. Přenos dat mezi přístupovými body je realizován pomocí bezdrátové technologie WiFi 802.11g, která je nejužším místem přenosové rychlosti v této úloze a to . Pro přenos signálu mezi zbývajících zařízeními je použita kroucená dvojlinka ve variantě UTP. V případě PC1 propojeným s přístupovým bodem 2 je k propojení využita křížená dvojlinka UTP.

PC2 je server a PC1 je klient.

Seznam použitých zařízení:

PC1, PC2	OS Windows 7 32-bit, OS Ubuntu 12.10
Směrovač 1, 2	Cisco Catalyst 2960 series
Přepínač 1, 2, 3	Cisco 2800 series
Přístupový bod 1, 2	Cisco 1242AG-E-K9

Obrázek 4 .22: Seznam zařízení topologie 3

7 Naměřené hodnoty

7.1 Topologie 1

Měření generováním TCP provozu

Analyzátory a generátory dat	Propustnost [Mb/s]
FETEST ParaScope GigE	98,7
Iperf	94,1
Thrulay	94,1
QCheck	90,9

Obrázek 4 .23: Generování TCP provozu

Měření generováním UDP provozu

Analyzátory a generátory dat	Generované přenosové pásmo [Mb/s]	Zpoždění [ms]	Variabilita zpoždění (Jitter) [ms]	Ztrátovost paketů [%]
FETEST ParaScope GigE	1	1,79	X	0
Iperf	1	X	0,016	0
Thrulay	1	1,15	0,025	0
QCheck	1	1,00	X	0

Obrázek 4 .24: Generování UDP provozu

*X – nástroj nepodporuje tento typ měření.

U všech naměřených hodnot jsme dosáhli srovnatelných výsledků, které se od sebe výrazně neodlišují. Popis konfigurace jednotlivých nástrojů pro jednotlivé testy naleznete v příloze.

7.2 Topologie 2

Měření generováním TCP provozu

Analyzátory a generátory dat	Propustnost [Mb/s]
FETEST ParaScope GigE	0,13
Iperf	0,12
Thrulay	0,11
QCheck	0,12

Obrázek 4 .25: Generování TCP provozu

Měření generováním UDP provozu

Analyzátory a generátory dat	Generované přenosové pásmo [Mb/s]	Zpoždění [ms]	Variabilita zpoždění (Jitter) [ms]	Ztrátovost paketů [%]
FETEST ParaScope GigE	1	4157	X	87,4
Iperf	1	X	3,13	88,0
Thrulay	1	5528	0	85,8
QCheck	1	4812	X	92,0

Obrázek 4 .26: Generování UDP provozu

*X – nástroj nepodporuje tento typ měření

U tohoto typu měření došlo generováním datového toku 1Mb/s vzhledem k velikosti propustnosti sériové linky stanovené na 0,1Mb/s k zahlcení fronty dat na směrovačích, a to se výrazně promítlo do vysokých hodnot zpoždění a ztrátovosti paketů. Také v tomto měření jsme u všech naměřených hodnot dosáhli srovnatelných výsledků, které se od sebe výrazně neodlišují. Popis konfigurace jednotlivých nástrojů pro jednotlivé testy naleznete v příloze.

7.3 Topologie 3

Měření generováním TCP provozu

Analyzátory a generátory dat	Propustnost [Mb/s]
FETEST ParaScope GigE	15,80
Iperf	14,20
Thrulay	13,89
QCheck	13,77

Obrázek 4 .27: Generování TCP provozu

Měření generováním UDP provozu

Analyzátory a generátory dat	Generované přenosové pásmo [Mb/s]	Zpoždění [ms]	Variabilita zpoždění (Jitter) [ms]	Ztrátovost paketů [%]
FETEST ParaScope GigE	1	2,76	X	0
Iperf	1	X	0,25	0
Thrulay	1	4,55	0.30	0
QCheck	1	2,00	X	0,01

Obrázek 4 .28: Generování UDP provozu

*X – nástroj nepodporuje tento typ měření

Při měření na topologii s bezdrátovým spojem o rychlosti 54Mb/s jsme dosáhli srovnatelných výsledků, které se od sebe výrazně neodlišují. Popis konfigurace jednotlivých nástrojů pro jednotlivé testy naleznete v příloze.

8 Zhodnocení

Měření probíhalo vždy ze směru klienta PC2 směrem k serveru PC1. Všechny testy měly dobu trvání 10 sekund a velikost přenášených paketů se pohybovala co nejbližší velikosti MTU.

Pro dodržení podmínek testování pro všechny generátory a analyzátory datového provozu byly stanoveny vlastnosti jako:

- Nastavení TCP protokolu pro měření propustnosti:

Maximální velikost segmentu (MSS) u testování propustnosti byla automaticky nastavena OS na nejvyšší možnou hodnotu vzhledem k MTU technologie Ethernet, která je 1500B pro data. Velikost MSS byla v průběhu testu u nástroje FETEST ParaScope GigE, Iperf, Thrulay a Qcheck 1448B. 1448B je výsledkem MTU 1500B – 52B značící velikost záhlaví IP protokolu 20B, záhlaví TCP protokolu 20B a volitelné nastavení 12B. Velikost TCP okénka (Window Size) byla ve všech měřeních nastavena shodně na 8192B, ověřeno nástrojem Wireshark.

- Nastavení UDP protokolu pro měření zpoždění, variability ve zpoždění a ztrátovosti paketů:

Velikost UDP datagramů byla nastavena na velikost 1460-1470B. Generované přenosové pásmo dosahovalo 1Mb/s.

V případě opakování tohoto měření se mohou jednotlivé výsledky naměřených hodnot lišit, avšak rozdíl by neměl být znatelný. Okolnostmi, které mohou ovlivnit naměřené hodnoty jsou výkon procesoru a jednotlivých komponent samotného počítače na straně klienta a serveru, tak i výběr prvků ležících na měřené trase jako například typ a délka použité kabeláže, tak i výkon jednotlivých směrovačů a přepínačů.

8.1 Použité nástroje

FETEST ParaScope GigE

- Klady

Ze všech testovaných nástrojů nabízí nejširší škálu využití. Jde o profesionální řešení s možností testování v terénu. Podpora všech druhů ethernetových portů do rychlosti 1Gb/s. Podpora generování více datových proudů, Test RFC2544, trasování, propustnost, zpoždění, délky kabelu, optického výkonu, BERT atd. Snadná přenositelnost vzhledem k velikosti a váze.

- Zápory

Neměří variabilitu zpoždění, cena.

Iperf

- Klady

Podpora pro mnoho OS. Měření lze provádět mezi PC s OS Linux a PC s OS Windows. Jednoduchá instalace. Iperf pro OS Windows nabízí grafické zobrazení, jednoduché nastavení a výstup hodnot v grafu. Ze softwarových nástrojů nejširší možnosti nastavení jako velikost přenášených dat, hodnota TTL, TCP okénka. Zahrnuta podpora IPv6 a testování v obou směrech najednou anebo v obou směrech individuálně. Možnost je také generovat více datových proudů. Ukládá výsledky do souboru. Běh serveru jako démon.

- Zápory

Neměří zpoždění a trasování.

Thrulay

- Klady

Měření všech testovaných parametrů v jednom nástroji. Jednoduché ovládání. Nastavení velikosti TTL, TCP okénka, času při testování.

- Zápory

Nepodporuje v OS Windows. Chybí podpora IPv6 a neukládá výsledky do souboru. Pro nastavení velikosti okénka nutno hodnoty zadávat jen v bajtech. Neměří trasování.

QCheck

- Klady

Snadná instalace a obsluha díky přehlednému a grafickému zpracování. Automatická volba jednotek pro naměřené hodnoty a u malých jednotek jako kb/s lze ocenit výpis na tři desetinná místa. Tento nástroj jako jediný z testovaných nabízí testování pro protokoly IPX a SPX používané v OS Novell Netware. Nabízí trasování.

- Zápory

Omezené měření propustnosti výrobcem nastaveno do 100Mb/s. Toto omezení se zřetelně promítlo při měření propustnosti topologie 1, kde bylo naměřeno 90,9Mb/s a v porovnání s ostatními výsledky v této topologii to byla nejnižší naměřená hodnota. Generovat lze jen jeden datový proud. U TCP protokolu omezení velikosti generovaných dat v rozsahu 1-1000 kB a u UDP protokolu vytížení linky v rozsahu 1-1000kb/s. Stáhnutí tohoto nástroje ze stránek společnosti Ixia je zdlouhavé. Nelze se obejít bez podpory MS Windows.

Neměří variabilitu zpoždění. Nastavení doby testování na maximální délku 10 sekund.

8.1.1 Shrnutí

Jako nejlepší pro měření propustnosti, zpoždění, variability zpoždění a ztrátovosti paketů se jeví nástroj Thrulay, který nabízí dostatečné možnosti nastavení, po přečtení manuálu lze považovat nastavení testování za velice intuitivní a také jako jediný nástroj ze všech testovaných nabídl výsledky pro všechny typy měření. Chybí jen podpora v OS Windows.

8.2 Přehled softwarových alternativ

Tabulka obsahuje přehled alternativ k nástrojům Iperf, Thrulay a Qcheck. Jedná se o volně dostupné alternativy.

Způsob výběru nástrojů probíhal na základě možnosti změřit alespoň dva ze základních testů pro měření vlastností sítí, jakými je propustnost, zpoždění, variabilita zpoždění a ztrátovost paketů.

Softwarové nástroje	Propustnost	Zpoždění	Variabilita zpoždění	Ztrátovost paketů	Vytížení PC procesoru	Pakety doručené mimo pořadí
Iperf	•		•	•		•
Thrulay	•	•	•	•		•
QCheck	•	•		•		
Nuttcp	•			•	•	
LMbench	•	•			•	
Uperf	•	•			•	

Obrázek 4 .29: Přehled softwarových alternativ

• - podpora daného typu testování.

9 Závěr

Hlavním bodem práce bylo vybrat tři volně dostupné alternativy generátorů a analyzátorů datového provozu, které by mohli konkurovat hardwarovému řešení FETEST ParaScope GigE. Zvolil jsem tedy, dle mého názoru, nejvhodnější volně dostupné nástroje a to Iperf, Thrulay a Qcheck. Provedl jsem názornou instalaci těchto nástrojů pod operačním systémem MS Windows a Linux. Dále následoval popis vlastností každého analyzátoru a generátoru datového provozu a tímto bodem jsem mohl již přejít k návrhu a samotné realizaci tří vhodných topologií, každou s jinými prvky, na kterých by bylo možné testování základních testů pro měření vlastností sítí. Postupně jsem změřil na každé ze tří topologií a s analyzátoru a generátory dat provedl měření propustnosti, zpoždění, variability zpoždění a ztrátovosti. Zjistil jsem, že při měření mi nejlépe vyhovuje měřicí nástroj Iperf, který poskytuje měření největší škálu možných nastavení při testování, avšak při mnou vybraných testech jsem musel zohlednit nástroj jako je QCheck, který disponuje omezeným nastavením parametrů. Pro práci v terénu mi z hodnocení vyšel nejlépe nástroj FETEST ParaScope GigE, který je malé velikost, hmotnosti a disponuje vlastní baterií. Jediným mnou shledaným nedostatkem FETESTU je nízký takt procesoru, který při náročnějších měřeních může čas strávený testováním prodloužit na dobu desítek minut. Thrulay je svým zpracováním velmi podobný Iperfu, podobně se i ovládá, a přestože nabízí méně možností pro nastavení testování, tak jako jediný z volně dostupných řešení dokáže podat výsledky o všech čtyřech vlastnostech sítí, které jsem testoval.

Osobním přínosem z vypracování tohoto projektu je pro mě přehled možných softwarových řešení a jejich vzájemné porovnání v možnostech testování. Taktéž jsem si v praktickém nasazení mohl vyzkoušet hardwarový generátor a analyzátor FETEST ParaScope GigE, který nabízí širokou škálu měření a využití. Také se prohloubil můj náhled do problematiky přenosu dat v počítačových sítích.

Z pohledu dalšího vývoje mi přijde zajímavé téma doplnění nástroje Thrulay o test vytížení procesoru při generování dat. Dále doplnění zobrazení propustnosti v grafu po nastavenou dobu testování a umožnění ukládání naměřených hodnot do souboru. Nástroj Thrulay je pro mě zajímavý hlavně tím, že jako jediný z testovaných nástrojů již splňuje testování propustnosti, zpoždění, variability zpoždění, ztrátovosti paketů a proto by jeho další rozšíření z něj udělalo všestranný nástroj.

Použitá literatura

- [1] ODOM, Wendell, CAVANAUGH, Michael. *Cisco QOS Exam Certification Guide*. Indianapolis: Cisco Press, 2004. ISBN 978-1-58720-124-2.
- [2] iperf.fr [online]: Tuning a UDP connection. Dostupné z: <http://iperf.fr/>
- [3] iperf.fr [online]: Tuning a TCP connection. Dostupné z: <http://iperf.fr/>
- [4] iperf.fr [online]: Iperf User Docs. Dostupné z: <http://iperf.fr/>
- [5] openmaniak.com [online]: Iperf. Dostupné z: <http://openmaniak.com/iperf.php>
- [6] e2epi.internet2.edu [online]: Thrulay: Manual page for thrulay. Dostupné z: <http://e2epi.internet2.edu/thrulay/thrulay.man.html>
- [7] sourceforge.net [online]: jperf 2.0.0. <http://sourceforge.net/projects/iperf/files/jperf/>
- [8] e2epi.internet2.edu [online]: thrulay, network capacity tester. <http://e2epi.internet2.edu/thrulay/>
- [9] kb.pert.geant.net [online]: Thrulay. <http://kb.pert.geant.net/PERTKB/ThrulayTool>
- [10] thrulay-ng.sourceforge.net [online]: thrulay-ng. <http://thrulay-ng.sourceforge.net/>
- [11] thrulay-hd.sourceforge.net [online]: thrulay-hd. <http://thrulay-hd.sourceforge.net/>
- [12] ixchariot.com [online]: QCheck Datasheet. <http://www.ixchariot.com/products/datasheets/qcheck.html>
- [13] practicallynetworked.com [online]: NetIQ's QCheck. <http://www.practicallynetworked.com/reviews/qcheck.htm>
- [14] COMER, D. E. *Internetworking with TCP/IP: Principles, Protocols, and Architecture 1* (5th ed.). Prentice Hall, 2006. ISBN 0-13-187671-6
- [15] FOROUZAN, BEHROUZ A. *Data Communications And Networking* (Fourth ed.). Boston: McGraw-Hill. pp. 621–630, 2007. ISBN 0-07-296775-7.
- [16] KAUFMANN, Morgan. *Deploying IP and MPLS QoS for Multiservice Networks: Theory and Practice* by John Evans, Clarence Filsfils, 2007. ISBN 0-12-370549-5.

Příloha

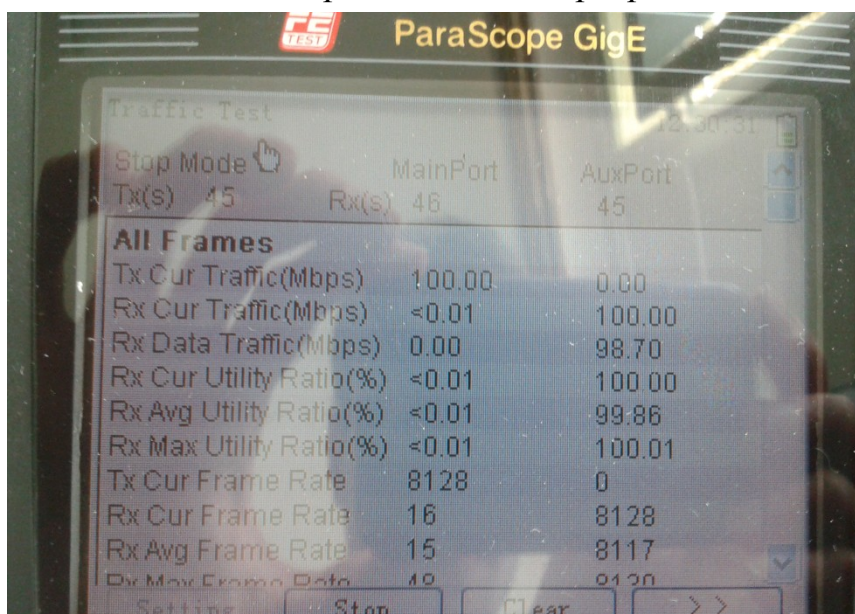
Příloha.A: Konfigurace a výstup jednotlivých testů.....I

Konfigurace a výstup jednotlivých testů:

- Topologie 1

FETEST ParaScope GigE

- *Generování TCP provozu, měření propustnosti*



Obrázek 5 .17: Měření propustnosti

- *Generování UDP provozu, měření zpoždění, ztrátovosti paketů:*

User Information

Local Main/Aux Port Cooperation

Test Layer IP Layer

Frame Length1500

Frame typeDIX TOS 1TTL 64

VLAN TagYes

Loopback DelayResult

Test Item	Result	Test Item	Result
Test Duration	10		
Tx (Mbps)	1.00	Rx(Mbps)	1.00
Tx Frames	854	Rx Frames	854
Error Frame	0	Avg. Delay	1.796
Max Delay	1.875	Min Delay	1.772

Obrázek 5 .18: Generování provozu

Vztah pro výpočet ztrátovosti paketů:

Iperf

○ *Generování TCP provozu:*

```
root@ladis-F5RL:/home/ladis# iperf -s -i 1 -f m
```

```
-----  
Server listening on TCP port 5001
```

```
TCP window size: 0.08 MByte (default)  
-----
```

```
[ 4] local 192.168.1.2 port 5001 connected with 192.168.2.2 port  
54926
```

```
[ ID] Interval Transfer Bandwidth
```

```
[ 4] 0.0- 1.0 sec 11.2 MBytes 94.1 Mbits/sec
```

```
[ 4] 1.0- 2.0 sec 11.2 MBytes 94.1 Mbits/sec
```

```
[ 4] 2.0- 3.0 sec 11.2 MBytes 94.2 Mbits/sec
```

```
[ 4] 3.0- 4.0 sec 11.2 MBytes 94.1 Mbits/sec
```

```
[ 4] 4.0- 5.0 sec 11.2 MBytes 94.1 Mbits/sec
```

```
[ 4] 5.0- 6.0 sec 11.2 MBytes 94.1 Mbits/sec
```

```
[ 4] 6.0- 7.0 sec 11.2 MBytes 94.1 Mbits/sec
```

```
[ 4] 7.0- 8.0 sec 11.2 MBytes 94.2 Mbits/sec
```

```
[ 4] 8.0- 9.0 sec 11.2 MBytes 94.1 Mbits/sec
```

```
[ 4] 9.0-10.0 sec 11.2 MBytes 94.1 Mbits/sec
```

```
[ 4] 0.0-10.2 sec 114 MBytes 94.1 Mbits/sec
```

○ *Generování UDP provozu:*

```
root@ladis-F5RL:/home/ladis# iperf -u -s -i 1 -f m
```

```
-----  
Server listening on UDP port 5001
```

```
Receiving 1470 byte datagrams
```

```
UDP buffer size: 0.16 MByte (default)  
-----
```

```
[ 3] local 192.168.1.2 port 5001 connected with 192.168.2.2 port  
41593
```

```
[ ID] Interval Transfer Bandwidth Jitter Lost/Total Datagrams
```

```
[ 3] 0.0- 1.0 sec 0.12 MBytes 1.00 Mbits/sec 0.012 ms 0/ 85 (0%)
```

```
[ 3] 1.0- 2.0 sec 0.12 MBytes 1.00 Mbits/sec 0.016 ms 0/ 85 (0%)
```

```
[ 3] 2.0- 3.0 sec 0.12 MBytes 1.00 Mbits/sec 0.015 ms 0/ 85 (0%)
```

```
[ 3] 3.0- 4.0 sec 0.12 MBytes 1.00 Mbits/sec 0.016 ms 0/ 85 (0%)
```

```
[ 3] 4.0- 5.0 sec 0.12 MBytes 1.00 Mbits/sec 0.013 ms 0/ 85 (0%)
```

```
[ 3] 5.0- 6.0 sec 0.12 MBytes 1.00 Mbits/sec 0.018 ms 0/ 85 (0%)
```

```
[ 3] 6.0- 7.0 sec 0.12 MBytes 1.00 Mbits/sec 0.017 ms 0/ 85 (0%)
```

```
[ 3] 7.0- 8.0 sec 0.12 MBytes 1.00 Mbits/sec 0.011 ms 0/ 85 (0%)
```

```
[ 3] 8.0- 9.0 sec 0.12 MBytes 1.00 Mbits/sec 0.016 ms 0/ 85 (0%)
```

```
[ 3] 9.0-10.0 sec 0.12 MBytes 1.00 Mbits/sec 0.016 ms 0/ 85 (0%)
```

```
[ 3] 0.0-10.0 sec 1.19 MBytes 1.00 Mbits/sec 0.016 ms 0/ 852 (0%)
```

Thrulay

○ *Generování TCP provozu:*

```
root@ladis-PC:/home/ladis# thrulay -t10 192.168.1.2
```

```
# local window = 262142B; remote window = 262142B
```

```
# block size = 8192B
```

```
# MTU: 1500B; MSS: 1448B; Topology guess: Ethernet/PPP
```

```
# MTU = getsockopt(IP_MTU); MSS = getsockopt(TCP_MAXSEG)
```

```
# test duration = 10s; reporting interval = 1s
```

```
 #(ID) begin, s end, s Mb/s RTT, ms: min avg max
```

```
( 0) 0.000 1.000 93.321 1.152 14.545 21.613
```

```
( 0) 1.000 2.000 94.439 9.948 14.816 19.736
```

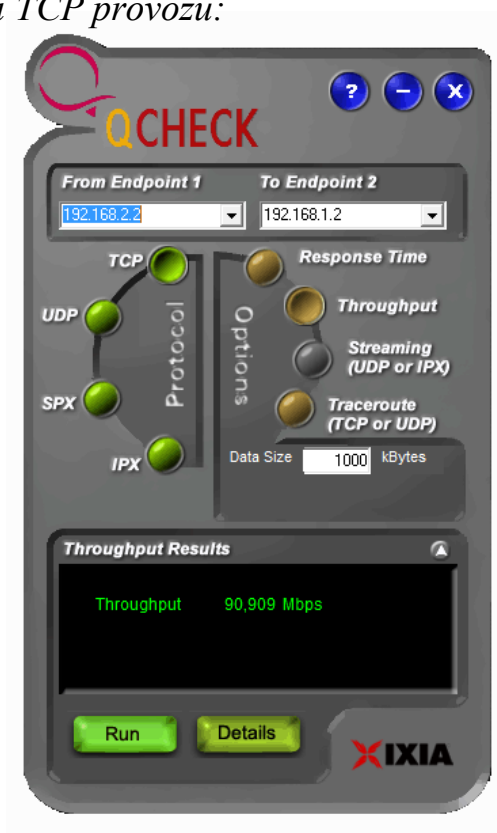
```
( 0) 2.000 3.001 94.124 9.854 14.840 24.354
( 0) 3.001 4.000 93.501 9.968 14.856 23.030
( 0) 4.000 5.000 94.376 9.973 14.904 25.017
( 0) 5.000 6.000 94.435 9.911 14.856 20.135
( 0) 6.000 7.000 94.043 9.884 14.893 23.841
( 0) 7.000 8.000 93.981 9.875 14.840 23.750
( 0) 8.000 9.000 93.583 9.927 14.839 19.489
( 0) 9.000 10.000 94.965 9.880 14.823 20.630
#( 0) 0.000 10.000 94.077 1.152 14.821 25.017
#(**) 0.000 10.000 94.077 1.152 14.821 25.017
```

○ *Generování UDP provozu:*

```
root@ladis-PC:/home/ladis# thrulay -t10 -u1M 192.168.1.2
Client runtime: 9.997s
Server used UDP buffer size of 262142 bytes
Client proposed sending 833 packets
Client said it sent 833 packets
Server received 833 total packets
Server received 833 unique packets
Delay: 19903.765ms
Delay quantiles (ignoring clock offset):
0th: 19903.728ms
25th: 19903.751ms
50th: 19903.765ms
75th: 19903.776ms
95th: 19903.794ms
Loss: 0.000%
Jitter: 0.025ms
Duplication: 0.000%
Reordering: 0.000%
no reordering
```

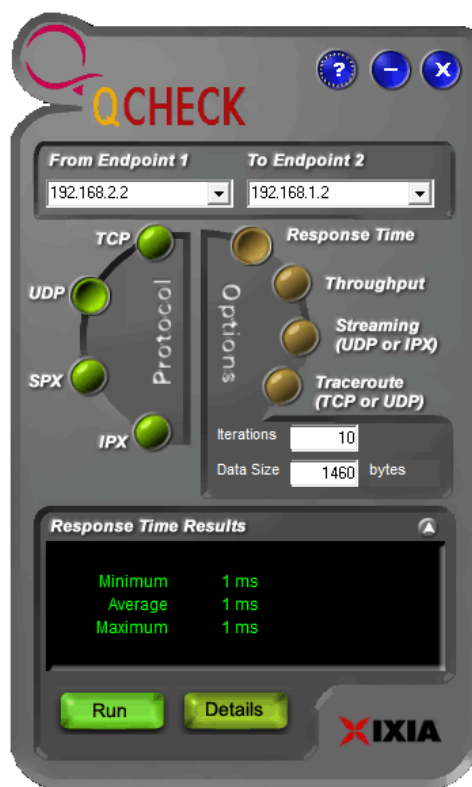
QCheck

- *Generování TCP provozu:*

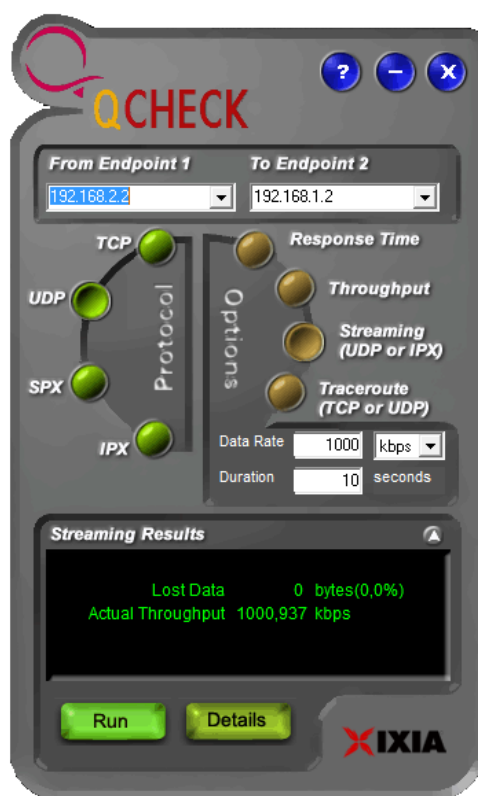


Obrázek 4 .30: *Propustnost*

- Generování UDP provozu:



Obrázek 4 .31: Zpoždění

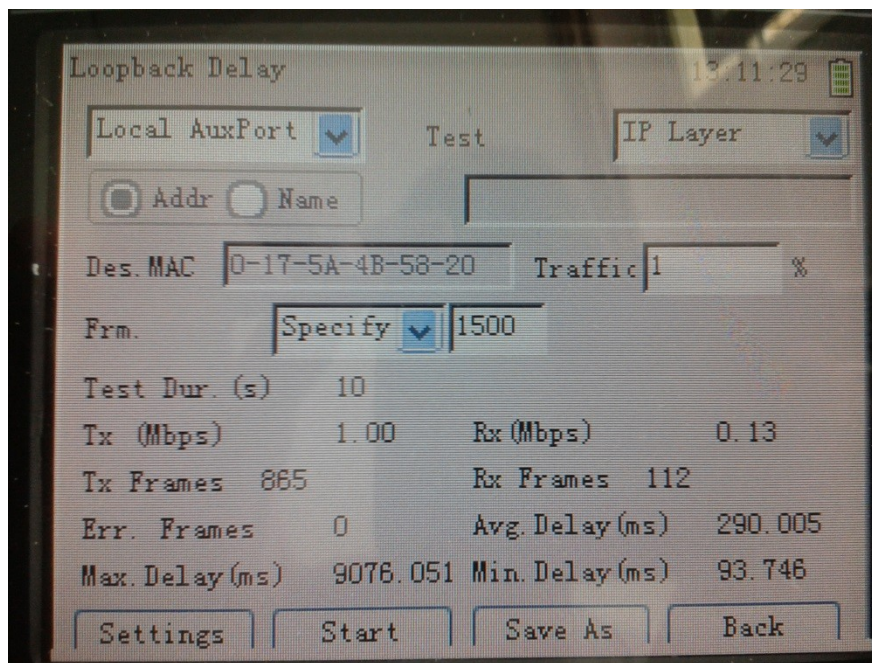


Obrázek 4 .32: Ztrátovost paketů

- Topologie 2

FETEST ParaScope GigE

- *Generování TCP provozu, měření propustnosti*



Obrázek 4.33: Měření propustnosti

- *Generování UDP provozu, měření zpoždění, ztrátovosti paketů:*

User Information

Local Main/Aux Port Cooperation

Test Layer IP Layer

Frame Length 1518

Frame type DIX TOS 1 TTL 64

VLAN Tag Yes

Loopback Delay Result

Test Item	Result	Test Item	Result
Test Duration	10		
Tx (Mbps)	1.00	Rx (Mbps)	0.12
Tx Frames	844	Rx Frames	107
Error Frame	0	Avg. Delay	4157.404
Max Delay	6379.924	Min Delay	98.217

Obrázek 5.19: Generování provozu

Vztah pro výpočet ztrátovosti paketů:

Iperf

○ *Generování TCP provozu:*

```
root@ladis-F5RL:/home/ladis# iperf -s -i 1 -f m
```

```
-----  
Server listening on TCP port 5001
```

```
TCP window size: 0.08 MByte (default)  
-----
```

```
[ 4] local 192.168.1.2 port 5001 connected with 192.168.2.2 port  
54927
```

```
[ ID] Interval Transfer Bandwidth
```

```
[ 4] 0.0- 1.0 sec 0.01 MBytes 0.12 Mbits/sec
```

```
[ 4] 1.0- 2.0 sec 0.02 MBytes 0.13 Mbits/sec
```

```
[ 4] 2.0- 3.0 sec 0.01 MBytes 0.12 Mbits/sec
```

```
[ 4] 3.0- 4.0 sec 0.02 MBytes 0.13 Mbits/sec
```

```
[ 4] 4.0- 5.0 sec 0.02 MBytes 0.13 Mbits/sec
```

```
[ 4] 5.0- 6.0 sec 0.01 MBytes 0.12 Mbits/sec
```

```
[ 4] 6.0- 7.0 sec 0.02 MBytes 0.13 Mbits/sec
```

```
[ 4] 7.0- 8.0 sec 0.01 MBytes 0.12 Mbits/sec
```

```
[ 4] 8.0- 9.0 sec 0.02 MBytes 0.13 Mbits/sec
```

```
[ 4] 9.0-10.0 sec 0.02 MBytes 0.12 Mbits/sec
```

```
[ 4] 10.0-11.0 sec 0.01 MBytes 0.12 Mbits/sec
```

```
[ 4] 11.0-12.0 sec 0.02 MBytes 0.13 Mbits/sec
```

```
[ 4] 12.0-13.0 sec 0.01 MBytes 0.12 Mbits/sec
```

```
[ 4] 13.0-14.0 sec 0.02 MBytes 0.13 Mbits/sec
```

```
[ 4] 14.0-15.0 sec 0.02 MBytes 0.13 Mbits/sec
```

```
[ 4] 15.0-16.0 sec 0.01 MBytes 0.12 Mbits/sec
```

```
[ 4] 16.0-17.0 sec 0.02 MBytes 0.13 Mbits/sec
```

```
[ 4] 17.0-18.0 sec 0.00 MBytes 0.03 Mbits/sec
```

```
[ 4] 18.0-19.0 sec 0.00 MBytes 0.00 Mbits/sec
```

```
[ 4] 19.0-20.0 sec 0.00 MBytes 0.00 Mbits/sec
```

```
[ 4] 20.0-21.0 sec 0.00 MBytes 0.00 Mbits/sec
```

```
[ 4] 21.0-22.0 sec 0.00 MBytes 0.00 Mbits/sec
```

```
[ 4] 22.0-23.0 sec 0.00 MBytes 0.00 Mbits/sec
```

```
[ 4] 23.0-24.0 sec 0.06 MBytes 0.46 Mbits/sec
[ 4] 24.0-25.0 sec 0.04 MBytes 0.37 Mbits/sec
[ 4] 0.0-25.6 sec 0.38 MBytes 0.12 Mbits/sec
```

○ *Generování UDP provozu:*

```
root@ladis-F5RL:/home/ladis# iperf -u -s -i 1 -f m
```

```
-----
Server listening on UDP port 5001
```

```
Receiving 1470 byte datagrams
```

```
UDP buffer size: 0.16 MByte (default)
-----
```

```
[ 3] local 192.168.1.2 port 5001 connected with 192.168.2.2 port
35205
```

```
[ ID] Interval Transfer Bandwidth Jitter Lost/Total Datagrams
```

```
[ 3] 0.0- 1.0 sec 0.01 MBytes 0.12 Mbits/sec 36.288 ms 0/ 10 (0%)
[ 3] 1.0- 2.0 sec 0.02 MBytes 0.13 Mbits/sec 59.708 ms 0/ 11 (0%)
[ 3] 2.0- 3.0 sec 0.01 MBytes 0.12 Mbits/sec 70.483 ms 0/ 10 (0%)
[ 3] 3.0- 4.0 sec 0.02 MBytes 0.13 Mbits/sec 76.521 ms 0/ 11 (0%)
[ 3] 4.0- 5.0 sec 0.02 MBytes 0.13 Mbits/sec 79.488 ms 0/ 11 (0%)
[ 3] 5.0- 6.0 sec 0.01 MBytes 0.12 Mbits/sec 80.865 ms 0/ 10 (0%)
[ 3] 6.0- 7.0 sec 0.02 MBytes 0.13 Mbits/sec 81.620 ms 0/ 11 (0%)
[ 3] 7.0- 8.0 sec 0.01 MBytes 0.12 Mbits/sec 44.700 ms 68/ 78 (87%)
[ 3] 8.0- 9.0 sec 0.02 MBytes 0.13 Mbits/sec 21.997 ms 77/ 88 (88%)
[ 3] 9.0-10.0 sec 0.02 MBytes 0.12 Mbits/sec 3.133 ms 77/ 88 (88%)
[ 3] 10.0-11.0 sec 0.01 MBytes 0.12 Mbits/sec 5.760 ms 70/ 80 (88%)
[ 3] 11.0-12.0 sec 0.02 MBytes 0.13 Mbits/sec 2.852 ms 77/ 88 (88%)
[ 3] 12.0-13.0 sec 0.02 MBytes 0.13 Mbits/sec 1.425 ms 77/ 88 (88%)
[ 3] 13.0-14.0 sec 0.01 MBytes 0.12 Mbits/sec 0.765 ms 70/ 80 (88%)
[ 3] 14.0-15.0 sec 0.02 MBytes 0.13 Mbits/sec 0.396 ms 77/ 88 (88%)
[ 3] 15.0-16.0 sec 0.01 MBytes 0.12 Mbits/sec 0.225 ms 70/ 80 (88%)
[ 3] 0.0-16.1 sec 0.24 MBytes 0.12 Mbits/sec 2.965 ms 681/ 852 (80%)
```

Thrulay

○ *Generování TCP provozu:*

```
root@ladis-PC:/home/ladis# thrulay -t10 192.168.1.2
# local window = 262142B; remote window = 262142B
# block size = 8192B
# MTU: 1500B; MSS: 1448B; Topology guess: Ethernet/PPP
# MTU = getsockopt(IP_MTU); MSS = getsockopt(TCP_MAXSEG)
# test duration = 10s; reporting interval = 1s
#(ID) begin, s end, s Mb/s RTT, ms: min avg max
( 0) 0.000 1.000 0.066 100.325 100.325 100.325
( 0) 1.000 2.000 0.066 1011.498 1011.498 1011.498
( 0) 2.000 3.000 0.262 2900.575 2900.584 2900.590
( 0) 3.000 4.000 0.000 0.000 0.000 0.000
( 0) 4.000 5.000 0.000 0.000 0.000 0.000
( 0) 5.000 6.000 0.328 5732.556 5732.571 5732.590
( 0) 6.000 7.000 0.000 0.000 0.000 0.000
( 0) 7.000 8.000 0.000 0.000 0.000 0.000
( 0) 8.000 9.000 0.000 0.000 0.000 0.000
( 0) 9.000 10.000 0.393 7926.855 8766.221 9125.523
#( 0) 0.000 10.000 0.111 100.325 5527.902 9125.523
#(**) 0.000 10.000 0.111 100.325 5527.902 9125.523
```

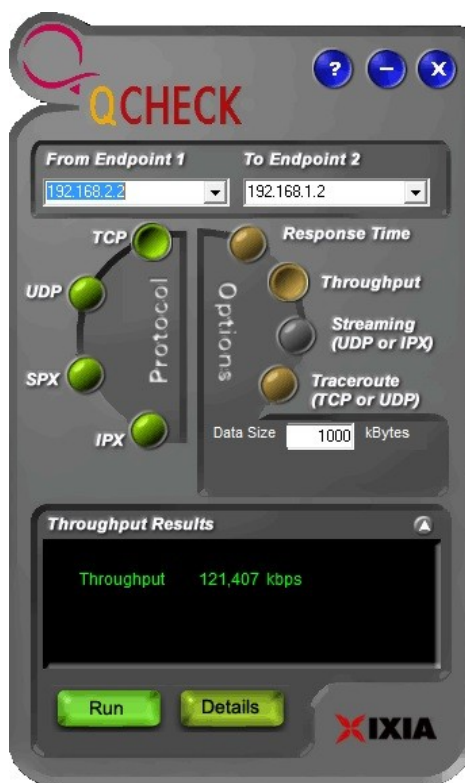
○ *Generování UDP provozu:*

```
root@ladis-PC:/home/ladis# thrulay -t10 -u1M 192.168.1.2
Client runtime: 10.016s
Server used UDP buffer size of 262142 bytes
Client proposed sending 833 packets
Client said it sent 833 packets
Server received 118 total packets
Server received 118 unique packets
Delay: infms
Delay quantiles (ignoring clock offset):
0th: 19479.781ms
```

25th: 21867.708ms
50th: 24335.802ms
75th: 25608.925ms
95th: 25615.831ms
Loss: 85.834%
Jitter: nanms
Duplication: 0.000%
Reordering: 0.000%
no reordering

QCheck

- *Generování TCP provozu:*



Obrázek 4.34: *Propustnost*

- *Generování UDP provozu:*



Obrázek 4 .35: Zpoždění

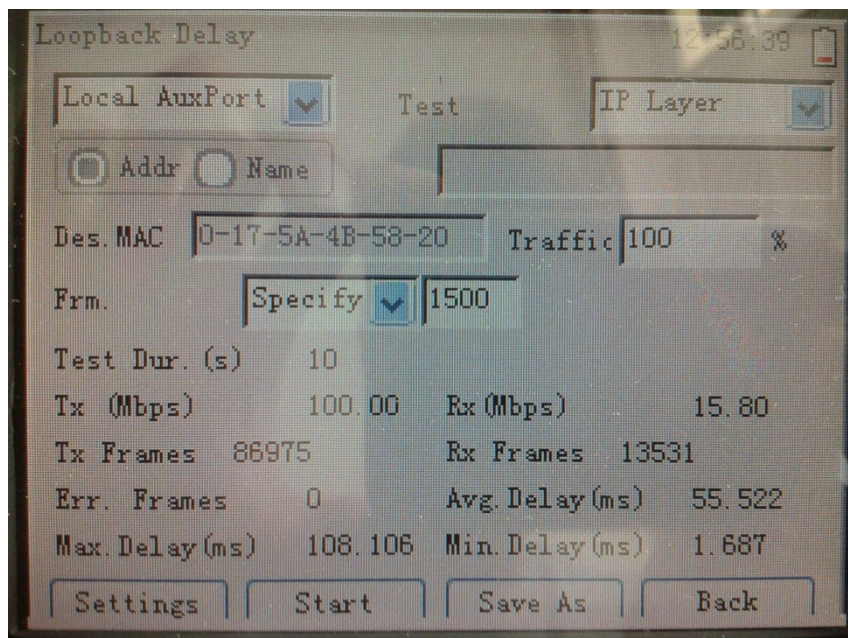


Obrázek 4 .36: Ztrátovost paketů

- Topologie 3

FETEST ParaScope GigE

- Generování TCP provozu, měření propustnosti



Obrázek 4 .37: Měření propustnosti

- Generování UDP provozu, měření zpoždění, ztrátovosti paketů:

User Information

Local Main/Aux Port Cooperation

Test Layer IP Layer

Frame Length 1500

Frame type DIX TOS 1 TTL 64

VLAN Tag Yes

Loopback Delay Result

Test Item	Result	Test Item	Result
Test Duration	10		
Tx (Mbps)	1.01	Rx (Mbps)	1.00
Tx Frames	872	Rx Frames	872
Error Frame	0	Avg. Delay	2.715
Max Delay	5.042	Min Delay	2.599

Obrázek 5 .20: Generování provozu

Vztah pro výpočet ztrátovosti paketů:

Iperf

○ *Generování TCP provozu:*

```
root@ladis-PC:/home/ladis# iperf -c 192.168.1.2 -m -i 1 -p 5001 -f m -t 10
```

```
-----  
Client connecting to 192.168.1.2, TCP port 5001
```

```
TCP window size: 0.02 MByte (default)  
-----
```

```
[ 3] local 192.168.2.2 port 51992 connected with 192.168.1.2 port 5001
```

```
[ ID] Interval Transfer Bandwidth
```

```
[ 3] 0.0- 1.0 sec 1.38 MBytes 11.5 Mbits/sec
```

```
[ 3] 1.0- 2.0 sec 1.25 MBytes 10.5 Mbits/sec
```

```
[ 3] 2.0- 3.0 sec 1.12 MBytes 9.44 Mbits/sec
```

```
[ 3] 3.0- 4.0 sec 1.25 MBytes 10.5 Mbits/sec
```

```
[ 3] 4.0- 5.0 sec 1.25 MBytes 10.5 Mbits/sec
```

```
[ 3] 5.0- 6.0 sec 1.50 MBytes 12.6 Mbits/sec
```

```
[ 3] 6.0- 7.0 sec 1.62 MBytes 13.6 Mbits/sec
```

```
[ 3] 7.0- 8.0 sec 1.50 MBytes 12.6 Mbits/sec
```

```
[ 3] 8.0- 9.0 sec 1.62 MBytes 13.6 Mbits/sec
```

```
[ 3] 9.0-10.0 sec 1.62 MBytes 13.6 Mbits/sec
```

```
[ 3] 0.0-10.1 sec 14.2 MBytes 11.8 Mbits/sec
```

```
[ 3] MSS size 1448 bytes (MTU 1500 bytes, ethernet)
```

○ *Generování UDP provozu:*

```
root@ladis-F5RL:/home/ladis# iperf -u -s -i 1 -f m
```

```
-----  
Server listening on UDP port 5001
```

```
Receiving 1470 byte datagrams
```

```
UDP buffer size: 0.16 MByte (default)  
-----
```

```
[ 3] local 192.168.1.2 port 5001 connected with 192.168.2.2 port 34036
```

```
[ ID] Interval Transfer Bandwidth Jitter Lost/Total Datagrams
[ 3] 0.0- 1.0 sec 0.12 MBytes 1.00 Mbits/sec 0.074 ms 0/ 85 (0%)
[ 3] 1.0- 2.0 sec 0.12 MBytes 1.00 Mbits/sec 0.108 ms 0/ 85 (0%)
[ 3] 2.0- 3.0 sec 0.12 MBytes 1.00 Mbits/sec 0.109 ms 0/ 85 (0%)
[ 3] 3.0- 4.0 sec 0.12 MBytes 1.00 Mbits/sec 0.120 ms 0/ 85 (0%)
[ 3] 4.0- 5.0 sec 0.12 MBytes 1.00 Mbits/sec 0.045 ms 0/ 85 (0%)
[ 3] 5.0- 6.0 sec 0.12 MBytes 1.00 Mbits/sec 0.184 ms 0/ 85 (0%)
[ 3] 6.0- 7.0 sec 0.12 MBytes 1.00 Mbits/sec 0.177 ms 0/ 85 (0%)
[ 3] 7.0- 8.0 sec 0.12 MBytes 1.00 Mbits/sec 0.073 ms 0/ 85 (0%)
[ 3] 8.0- 9.0 sec 0.12 MBytes 1.00 Mbits/sec 0.126 ms 0/ 85 (0%)
[ 3] 9.0-10.0 sec 0.12 MBytes 1.00 Mbits/sec 0.256 ms 0/ 85 (0%)
[ 3] 0.0-10.0 sec 1.19 MBytes 1.00 Mbits/sec 0.245 ms 0/ 852 (0%)
```

Thrulay

o *Generování TCP provozu:*

```
root@ladis-PC:/home/ladis# thrulay -t10 192.168.1.2
# local window = 262142B; remote window = 262142B
# block size = 8192B
# MTU: 1500B; MSS: 1448B; Topology guess: Ethernet/PPP
# MTU = getsockopt(IP_MTU); MSS = getsockopt(TCP_MAXSEG)
# test duration = 10s; reporting interval = 1s
#(ID) begin, s end, s Mb/s RTT, ms: min avg max
( 0) 0.000 1.000 14.483 4.548 75.886 127.583
( 0) 1.000 2.000 14.353 47.655 87.978 140.805
( 0) 2.000 3.000 13.304 55.724 95.244 162.965
( 0) 3.000 4.000 13.565 47.381 90.905 141.113
( 0) 4.000 5.000 13.828 53.698 88.617 130.159
( 0) 5.000 6.000 14.090 52.701 89.005 131.864
( 0) 6.000 7.000 14.680 41.733 91.316 155.110
( 0) 7.000 8.000 12.714 58.507 96.631 150.884
( 0) 8.000 9.000 13.894 57.752 89.713 129.800
( 0) 9.000 10.000 13.959 41.737 89.284 175.701
#( 0) 0.000 10.000 13.887 4.548 89.317 175.701
#(**) 0.000 10.000 13.887 4.548 89.317 175.701
```

- *Generování UDP provozu:*

```
root@ladis-PC:/home/ladis# thrulay -t10 -u1M 192.168.1.2
```

Client runtime: 9.999s

Delay: 342.483ms

Loss: 0.000%

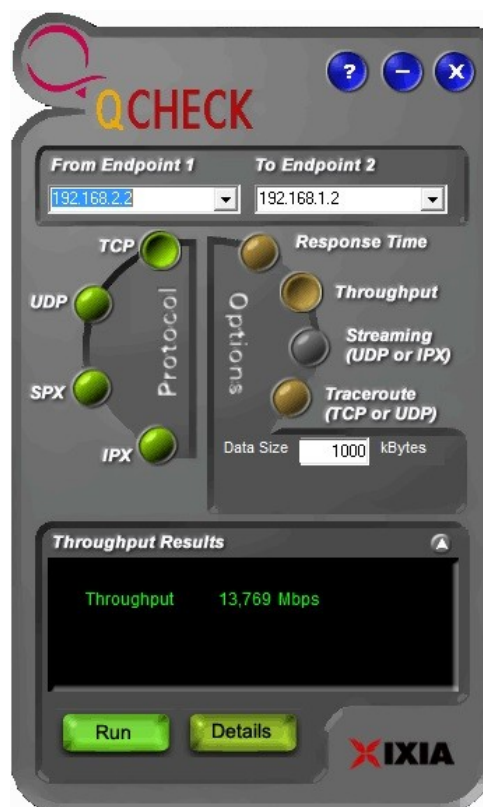
Jitter: 0.296ms

Duplication: 0.000%

Reordering: 0.000%

QCheck

- *Generování TCP provozu:*

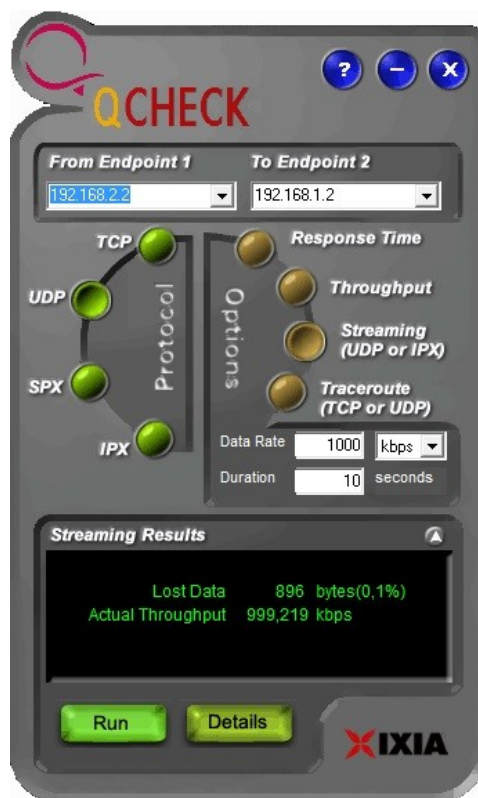


Obrázek 4.38: Propustnost

- Generování UDP provozu:



Obrázek 4 .39: Zpoždění



Obrázek 4 .40: Ztrátovost paketů